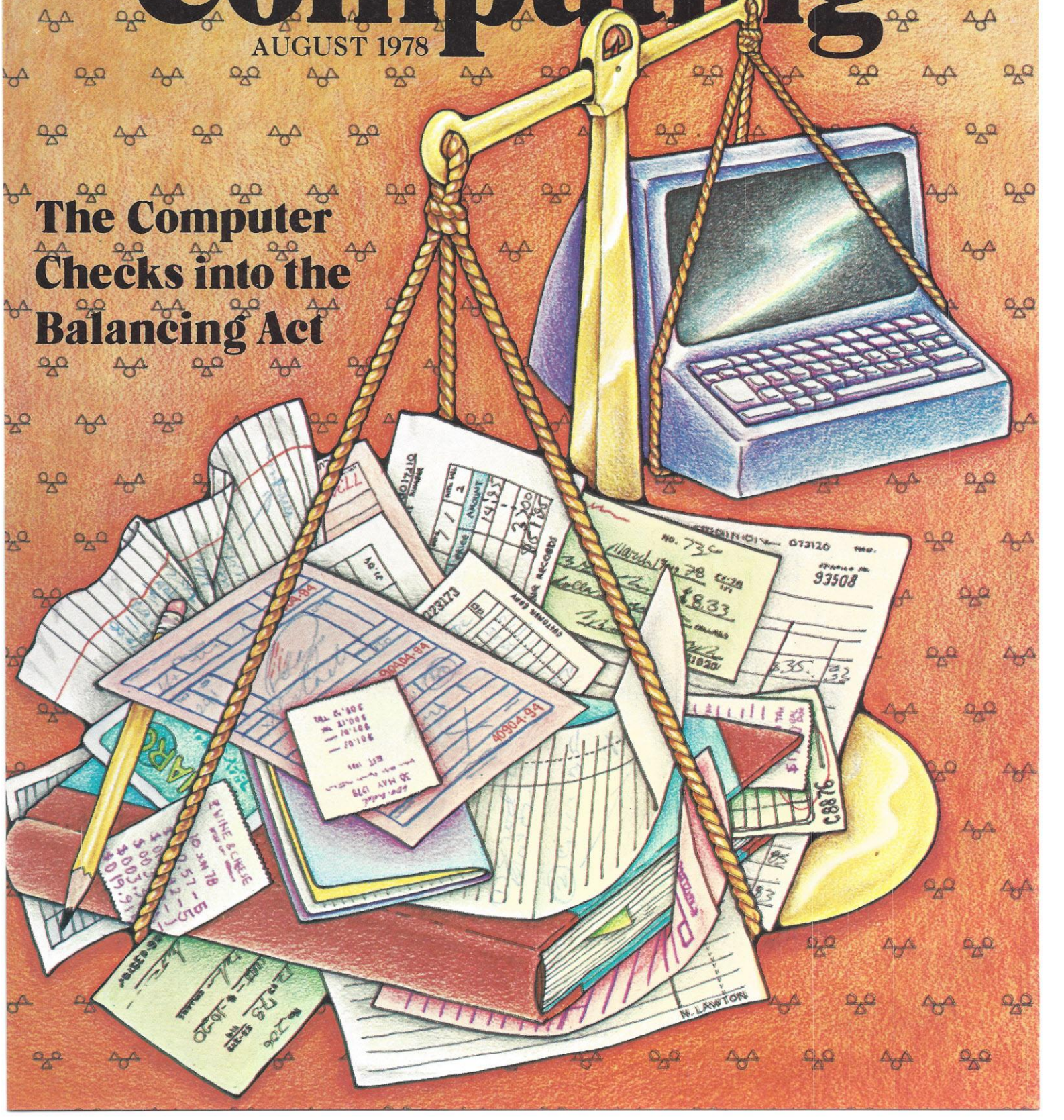
Building a Heathkit H-8 • TRS-80 • Contract Labor • Bartending

$2.00

Personal Computing

AUGUST 1978

The Computer Checks into the Balancing Act



INTRODUCING THE G/2 PROGRAM LIBRARY.

Now you don't have to be a programmer to use your computer!

At last there's a reliable, professional source of useful and entertaining programs for your personal computer. The G/2 Program Library from GRT unlocks the full power of your personal computer. Without any programming knowledge on your part.

The G/2 Program Library consists of applications programs pre-recorded on cassette tapes. A detailed manual with each cassette provides complete instructions. And G/2 programs contain prompting instructions. So they're easy to use every step of the way. Just load the tape and G/2 puts your computer to work.

The first three G/2 program tapes are available now: For exciting family games, get **Beat the House.** You'll find Blackjack, Craps, Roulette and Slot Machine programs, with truly random chances and realistic, casino-like payoffs.

Interested in your health? You'll want **Clinic.** Biorhythms, Dieting and Longevity programs give you new perspectives in health and happiness, along with some pleasant guidelines to follow.

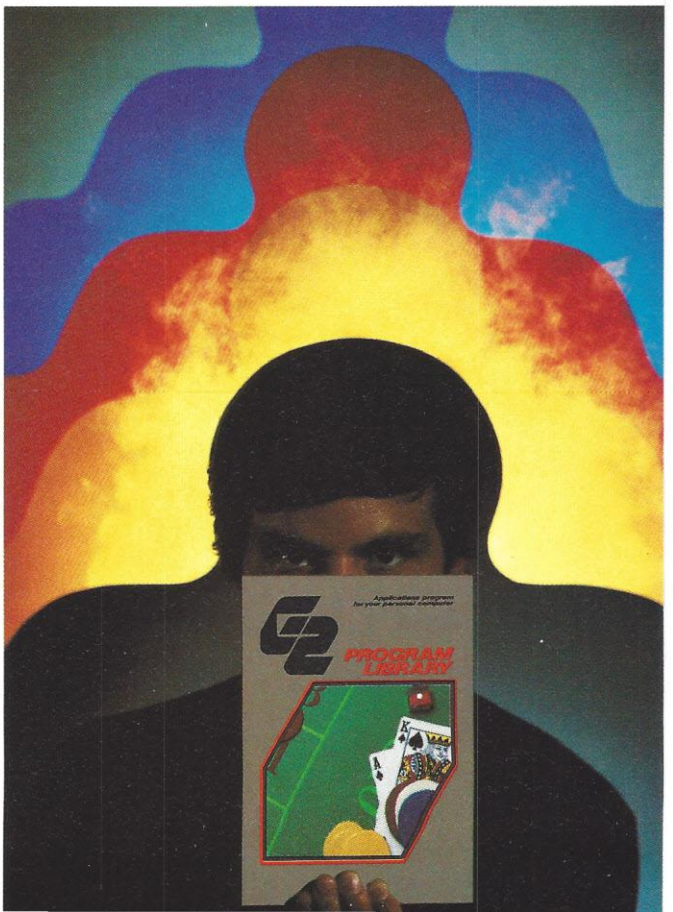
Need a better handle on your money? Buy **Dollars & Sense.** Checkbook is a comprehensive program to balance your account and keep within your budget, while Best Choice is a novel way to make decisions based on the facts you know and the relative importance of each. Use it for everything from choosing a new car to selecting your next stereo.

Soon, every dealer that handles personal computers will stock the G/2 Program Library. For the G/2 dealer nearest you, call us toll-free at 800/538-1770. (In California, 800/662-9810.) And if your favorite computer dealer doesn't yet offer G/2, have him call the same number.

Soon you'll see G/2 program tapes in many other fields of interest. The G/2 Program Library is now available for use with SOL, SWTPC, Exidy, TRS-80 and Apple personal computers. Just select the package marked for the computer you own.

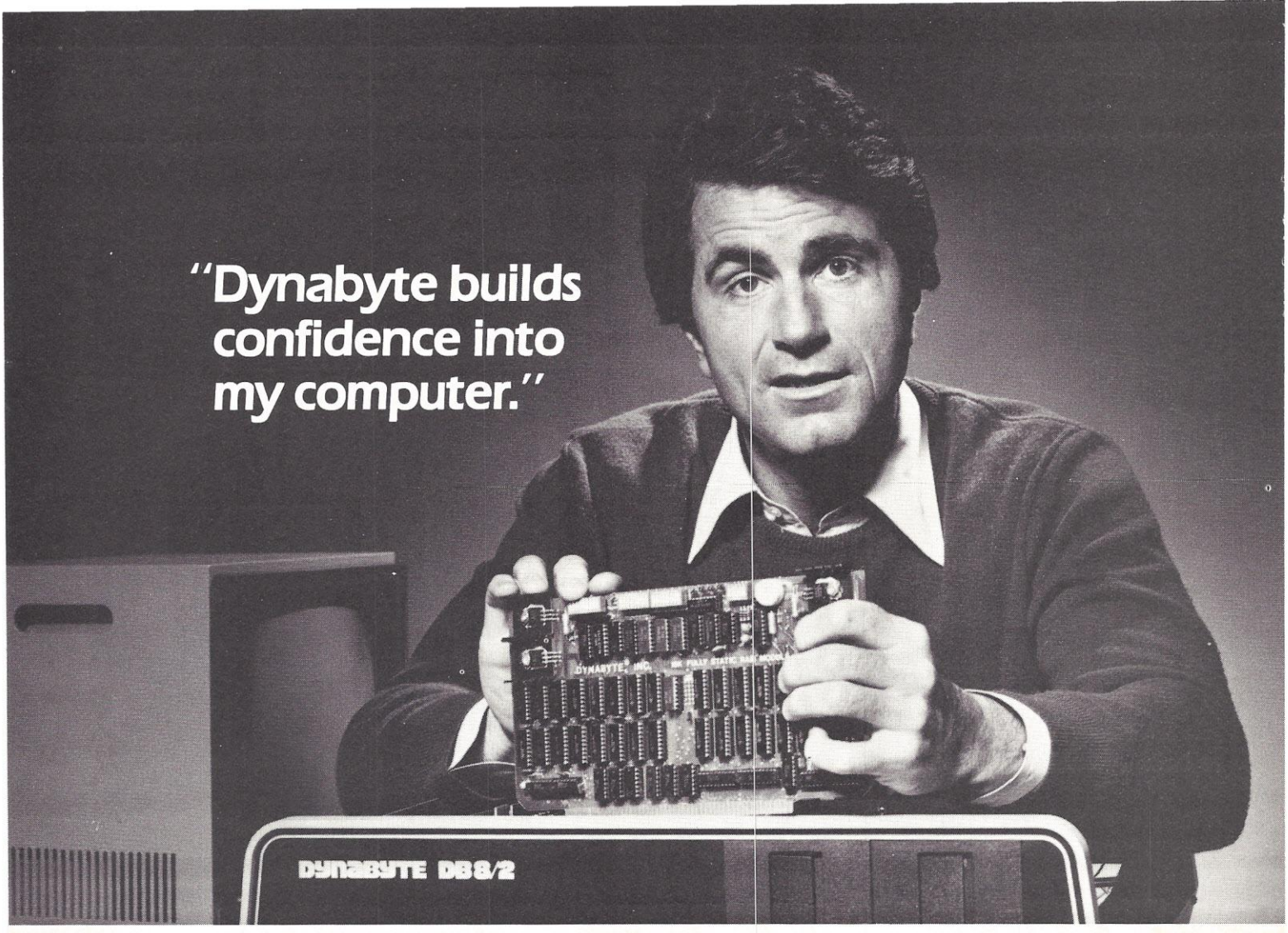
Start building your G/2 Program Library now. And put the full power of your computer to work for you.

THE REASON YOU BOUGHT YOUR COMPUTER. G/2™



GRT® CORPORATION

A product of GRT Corporation
Consumer Computer Group
1286 North Lawrence Station Road
Sunnyvale, California 94086
(408) 734-2910



Plug-in-and-run performance takes the uncertainty out of owning a computer. That's the kind of computer products Dynabyte builds.

Confidence in your computer is increased because you know that Dynabyte S-100 boards are factory assembled by trained technicians, burned in for 72 hours at elevated temperatures and continuously tested for three days.

That was true of the industry's first assembled and tested 16k dynamic RAM from Dynabyte, and it continues with all the other products from Dynabyte.

Once you plug Dynabyte boards in their careful design, factory prime components and quality construction keep them running. Dynabyte backs them with a one year warranty—the longest in the industry.

Incompatability is coming to an end. Dynabyte's 16k and 32k static RAMs come with access times of either 250ns for Z-80A processors or 450ns for 8080 and 8085 chips. You can use the 16k static RAMs with Alpha Micro or Cromemco bank switching memory expansion techniques.

Computer users are so confident of Dynabyte's 16k dynamic RAM they have made it the most widely used S-100 dynamic memory in the world.

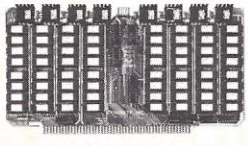
If you intend to become a proficient data processor but instead are unhappily debugging your system, or want to avoid a lot of debugging, then Dynabyte will be a great choice.

As you spend less time monkeying with hardware and more time computing, you'll become more confident of your abilities and proud of your accomplishments. Isn't that what owning a computer is all about?

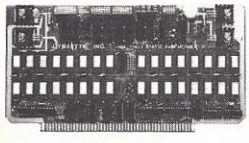
Tell your computer retailer with confidence that you want Dynabyte boards—the plug-in-and-run performers.

1005 Elwell Court,
Palo Alto, CA 94303.
(415) 965-1010.

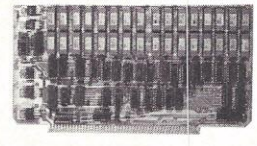
DYNABYTE



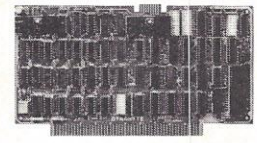
32k Fully Static RAM. 250ns or 450ns; 4k boundary addressing; no DMA restrictions; full Schmitt trigger buffering; no wait states; complete S-100 compatibility; conservative thermal design includes eight regulators and heat sinks. 250ns, $995. 450ns, $925.



16k Fully Static RAM. 250ns or 450ns; bank select is provided and is compatible with most popular schemes for memory expansion beyond 64k including Alpha Micro and Cromemco systems; 4k block addressing along 4k boundaries; write protect with alarm for each 4k block; full Schmitt trigger buffering. 250ns, $555. 450ns, $525.



16k Dynamic RAM. The industry's most popular S-100 dynamic RAM. Self-contained refresh logic is transparent to the 8080 processor and never generates a WAIT state. 1MHz direct memory access, 16k addressing. The Original Great Memory by Dynabyte, $399.

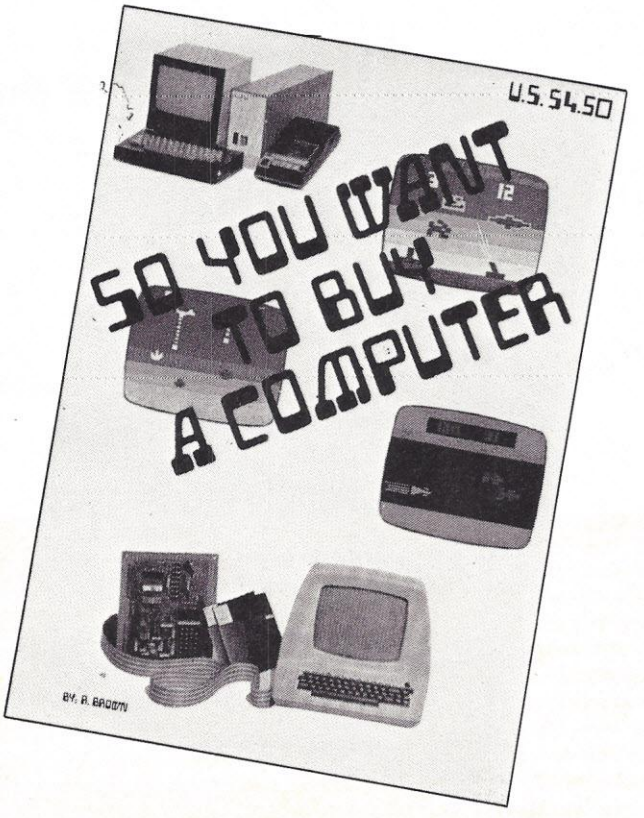


Naked Terminal. 80 character × 24 line = 1920 character display capacity. Not just a video interface but a full terminal; plug into a mainframe with keyboard and monitor and save! Upper and lower case; black/white or white/black; block mode editing; addressable cursor; half and full duplex; no software support required; addressable status and I/O ports. $350.

CIRCLE 4

GET THE REAL LOW DOWN BEFORE YOU BUY

SO YOU WANT TO BUY A COMPUTER??



This new book presents an objective look at the top 24 micro systems sold throughout the world. It discusses the Pros and Cons of each system in No Uncertain Terms and takes a straight-forward look at the micro computer industry as it relates to YOU. Written especially for the layman in a language he can understand. Profit from the mistakes of others.

Includes hundreds of references.

Table of Contents

1. Introduction
2. Don't get hung up on the chips
3. Which category do you fit into
4. Now – About the Hardware
5. Peripherials that plug in?
6. What? No Software!
7. Helpful Suggestions before spending money
8. Addresses

$4.50 per Volume
Volume discounts available

SCIENTIFIC RESEARCH

P.O. Box 49099-P
Key Biscayne, Florida 33149

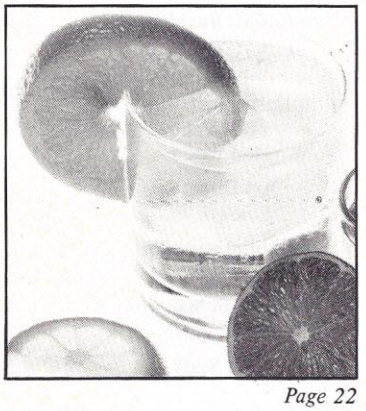
Phone Orders Call 800-327-6543

Add $1 handling, all domestic shipments sent U.P.S. except APO and P.O. Box which go parcel post. Foreign orders add $4 for air shipment and make payable in U.S. funds.

CIRCLE 5

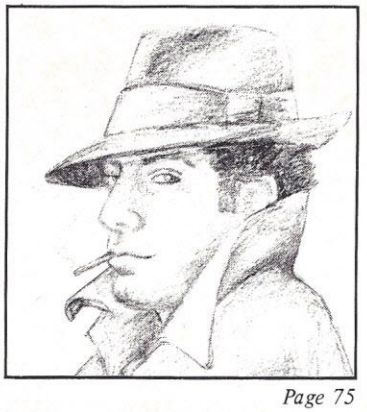
Personal Computing

AUGUST 1978 VOLUME II, NO. 8



Page 22

Page 37



Page 75

Cover illustration by Nancy Lawton

DEPARTMENTS

LAUNCHING PAD

DIGGING IN

IN THE MONEY

ON THE LIGHTER SIDE

Personal Computing

AUGUST 1978 VOL. II, NO. 8

Publisher/Editor
Harold G. Buchbinder

Assistant Editors
Gary Dozier
Harry Shershow
Ellen Stein
Don Wood

Art Director
Jane Higgins

Editorial Assistant
Marjorie Morse

Production Manager
Charles Verret

Art Staff
Michael Barisano
Stephen Fischer
Barbara Leonard
Richard Sarno

Editorial Production
Sarah Jewler
Wendy Michael

Advertising Production
Joe Ingram

Composition
Peter Bochner
Julianna Crosier
Kathy Gagne
Cheryl Kroll

Circulation
Regina Harrington
Dave Harriman

General Administration
Jonna Yager
Esther Shershow
Marion Pearlman
Sarah Binder
Don Schaaf

Advertising Sales. Northeast: George Palken, Bill Mahan, 1050 Commonwealth Ave., Boston, MA 02215; (617) 232-5470 • **Mid-Atlantic:** Arthur Daks, Benwill Publishing Corp., 69 So. Central Ave., Valley Stream, NY 11580; (516) 872-9550 • **Mid-West:** Hank Bean, 2633 Hillside Lane, Evanston, IL 60201; (312) 475-7173 • **Northwest:** Ralph Petersen, 1380 Oak Creek Drive, Palo Alto, CA 94304; (415) 328-4392 • **Southwest:** Yuri Spiro, Carol Stagg, Benwill Publishing Corp., 924 Westwood Blvd., Los Angeles, CA 90024; (214) 478-3017 • **Japan:** Hiro H. Irie, International Business Corp., 11-8, Narita-Higashi 1-chome, Suginami-ku, Tokyo 166: Phone (03) 311-1746

Subscription rates. U.S.: 1 year (12 issues) $14; 2 years (24 issues) $26; 3 years (36 issues) $38. Canada & Mexico: add $4/year for surface mail, $8/year for airmail. In Japan/Asia: Personal Computing, c/o CQ Publishing Co., 14-2 Sugamo 1-chome, Toshima-ku, Tokyo 170, Japan. All other countries: add $8/year for surface mail, $36/year for airmail. Send subscription orders and changes of address to: Circulation, Personal Computing, 1050 Commonwealth Ave., Boston, MA 02215. **Back issues.** U.S.: $3. Canada & Mexico: $4. All other countries: $6.

Publisher. Published monthly by Benwill Publishing Corp.: Harold G. Buchbinder, Chairman of the Board; George Palken, President; Esther Shershow, Treasurer. Executive, Editorial and Subscription Offices: 1050 Commonwealth Ave., Boston, MA 02215. Controlled Circulation postage paid at Long Prairie, MN. Membership in Audit Bureau of Circulation pending.

FEEDBACK

Walloping Response

Dear Editor:

I really enjoyed the article by Herbert Schildt, "Wall $treet Wallop", in the May issue.

I have been hoping someone could get me started using my computer in this area, and it just filled the bill!

I hope you publish more articles in this area, including programs as in "Wall $treet Wallop". Loved it and your magazine.

Thoras Thornberg
Champaign, IL

Editor's note: While some people use their computers for purely business purposes and others prefer computer games and recreations, most mix business and pleasure. (Here at Personal Computing, *we like to play Star Trek after hours on our computer.) So we present a balance of articles to appeal to all tastes.*

* * *

Dear Editors:

Loved "Wall $treet Wallop"! I am looking forward to many more articles along these lines.

The program was excellent. My hat's off to the author.

Bob Verket
Bloomington, IL

* * *

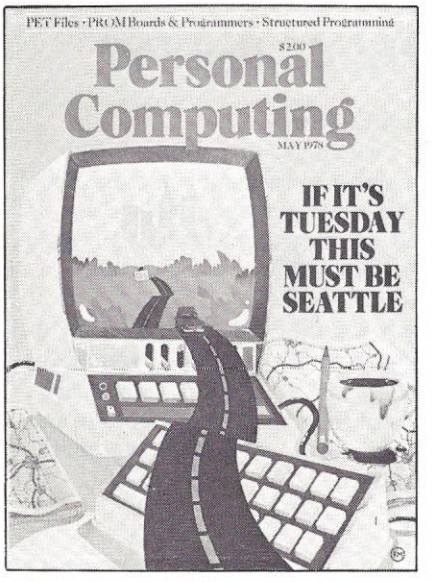
Dear Editors:

"Wall $treet Wallop" (May 1978) was really a useful article. It should improve my investment success immediately. The author suggests that the correlation between the price of a stock and the DJI average be calculated and the results used to predict the future price of the stock with some unmentioned level of confidence. Now all I need to know is the high and low of the DJI average this year and next year.

Rob Cave
Irving, TX

Author's note: Glad you enjoyed it, Rob. As soon as I dust off my crystal ball I'd be glad to let you know the future! But for the time being, we will just have to use our intuition.

Actually, what you are suggesting is a time series analysis based on the DJI *average. I have been working on just*



such a program, but for those of you who might be in a hurry to "predict" the movements of the Dow, I refer you to a very easy-to-understand and straightforward book on statistics by Harnett and Murphy, Introductory Statistical Analysis. *Also, it only takes five lines of code (5) to add a "confidence factor" to the program given in the article, but I had reached the end of my RAM, and left it out. However, the confidence interval is directly derived from the coefficient of determination, etc. Therefore, those with more RAM can easily add this feature.*

As a last word, I would be very interested to see other articles on statistical market analysis; it is such a rich and engaging area.

—Herbert Schildt

Pet Strings

Dear Editors:

This letter is in response to Karl & Susan Quosig's (April 1978) request for a routine to utilize the statements "CHANGE S$ TO S" and/or "CHANGE S TO S$" on their PET computer.

"CHANGE S$ TO S" places the numeric representation of the ASCII character into an array S with element S(0) = the length of string S$. The following routine will work on the PET, giving the same result that a CHANGE statement would (e.g., "A" will be converted to 65).

```
10 DIM S(80)
15 INPUT S$
20 S(0) = LEN(S$)
25 PRINT S(0)
30 FOR X = 1 TO LEN(S$)
40 S(X) = ASC(MID$(S$,X,1))
45 PRINT S(X)
50 NEXT X
```

"CHANGE S TO S$" converts the numeric representation of the ASCII character (e.g., 65 will be converted to the character "A"). Element S(0) contains the length of the string. Use the following routine for the PET computer:

```
10 DIM S(80)
15 INPUT S(0)
25 FOR Y = 1 TO S(0)
35 INPUT S(Y)
45 NEXT Y
50 FOR X = 1 TO S(0)
60 B$ = CHR$(S(X))
70 S$ = S$ + B$
80 NEXT X
90 PRINT S$
```

Good luck!

Darryl M. Maloof
Los Angeles, CA

* * *

Dear Editor:

This letter is in response to Karl and Susan Quosig's inquiry which appeared in the April 1978 issue of *Personal Computing.*

The "CHANGE" statement permits the user to transform a character string into a list of numeric values, or a list of numeric values into a character string. Each ASCII character in a string can be converted to its ASCII numeric (decimal) equivalent value, and vice versa. As an illustration, consider the following two examples:

```
 5 REM * EXAMPLE ONE
10 DIM A(3)
20 LET A$ = "DOG"
22 PRINT A$
30 CHANGE A$ TO A
40 PRINT A(0);A(1);A(2);A(3)
50 END
RUN
DOG
 3    68    79    71
READY
```

```
 5 REM * EXAMPLE TWO
10 DIM A(3)
20 LET A(0) = 3
21 LET A(1) = 68
22 LET A(2) = 79
23 LET A(3) = 71
24 FOR N = 0 TO 3
25 PRINT A(N);
26 NEXT N
30 CHANGE A TO A$
40 PRINT A$
50 END
RUN
 3   68   79   71
DOG
READY
```

In example one, the character string "DOG" was converted to its numeric values which were stored in subscripts 1, 2 and 3 of list A. The length of the string was stored in A(0). In example two, the numeric values stored in subscripts 1, 2, and 3 of list A were converted back to their equivalent ASCII characters. (A(0) was not converted.)

To run these programs (or similar ones) on your PET 2001, make the following changes by removing line 30 from both programs and inserting these routines:

```
25 REM * SUBSTITUTION
         ROUTINE FOR
26 REM   CHANGE A$ TO A
27 LET I = LEN(A$)
28 LET A(0) = I
29 FOR J = 1 TO I
30 LET B$ = MID$(A$,J,1)
31 LET A(J) = VAL(B$)
32 NEXT J
```

```
27 REM * SUBSTITUTION
         ROUTINE FOR
28 REM   CHANGE A TO A$
29 FOR J = 1 TO A(0)
30 LET I = A(J)
31 LET B$(J) = STR$(I)
32 LET A$ = B$(J-1) + B$(J)
33 NEXT J
```

Bob Connors
Columbia, SC

PROM Pen Pal

Dear Editors:

I would like to correspond with anyone who has built a microsystem from scratch, using pre-etched boards and self-programmed PROMs. I'd also like information on manual programming of PROMs.

In return, I will supply answers to anyone having questions about a BASIC or RPG II program. Also, I have formal training and experience in accounting and can be of assistance in that area.

Bruce Showalter
857 Cedar
Abilene, TX 79601

Copyrighting Programs

Dear Editors:

Many published programs contain a copyright REM statement. What are the procedures necessary in order to copyright a program for personal use, for publishing, or for selling? What are the costs involved?

William J. Beyda
Rockville Centre, NY

* * *

Dear Editors:

I have enjoyed many of the articles in *Personal Computing* but I think that an excellent topic has been neglected. I refer to the question of law and ethics with regard to the handling of software and specifically the question of plagiarism.

What does the law say about plagiarism and what are the ethical responsibilities of software users, including hobbyists, businessmen who are end users, systems designers and software producers?

At what point is copying socially acceptable and when do we cross over the line into the reprehensible? Obviously, the hobbyist can copy a program listing from *Personal Computing* and put it in his computer. Furthermore, he can save it on a cassette to use again. (After all, if you didn't want it copied you wouldn't have put it in the book.) But is the hobbyist guilty of anything if he makes another copy for his friend down the street and gives it to him? What if he sells it to him? What if the original published program was written in Hewlett-Packard BASIC and the hobbyist copies it for a TRS-80 so that extensive revision is necessary?

Say you are an end user. Can you copy a program from a book and use it in your business? Can you copy it 30 times for a multibranch organization? Can you or should you copy a tape from a licensed distributor of software and use it in a multibranch operation?

What if you run a programming service? You use the book, but change the statements to create a different format in the printout. Is this okay? If not, what would it take to make it all right? Would presenting your customer with another copy of the same book square things or must the penance be deeper?

Obviously software companies take the matter seriously because most of the software companies have a licensing agreement for you to sign before you make a purchase. But what legally is the situation? And perhaps even more important, what is the situation ethically? Most of the violations would cause trivial losses on an individual basis but overall could deny an innovator the fruits of his brainstorm. In the long run, this would not benefit the hobby.

I wish someone would write an article on this subject to let us all know where we stand.

James H. Sheats
East Point, GA

Editor's note: Any article, including programs, art, and so forth, which is submitted to a magazine is copyrighted upon publication, sparing the author that task. All programs, therefore, appearing in Personal Computing *are protected by copyright law. They are meant to be used by the reader for his or her own personal interests. However, copyright law forbids reselling someone else's property for profit without prior permission from the copyright holder. The law also prohibits rewriting a copyrighted article into another language for resale. It would be wise to consult a lawyer before marketing a product which*

resembles a similar product already on the market.

The routine for obtaining a copyright to an unpublished work is fairly simple. Write to the Copyright Office, Library of Congress, Washington, D.C. 20559 and request application form TX. When the application for copyright is received, fill it out, enclose one unpublished copy of your material together with $10 and send it in. That's all there is to it. Answers to further questions regarding copyright, including most recent revisions, will be found in the pamphlet which the Copyright Office will also send to you.

Praise for PC

Dear Editors:

I have just read (and re-read) your March 1978 issue, which I purchased off the newsstand. Your magazine is very informative and useful for novices like me. This was my first computer magazine, and I found several articles to be "readable" and informative, including, "Computer Languages: Tools of the Trade", "Diving into Computer Advertising", "Facts on Floppies", and your department article, "Future Computing".

Fred J. Quinn, Jr.
Waterford, MI

Pet and PC Delivery

Editors:

Ref. "Pet Dealings" *PC* Feedback May '78.

PET order vs. delivery

3-15-78 LD Telcon Ft. Worth, TX to Commodore Palo Alto, CA requesting PET brochure
3-20-78 Brochure received in mail
3-21-78 $795 check mailed for one PET 2001-8
3-23-78 Commodore typed invoice
3-24-78 Commodore typed invoice
3-27-78 Copy of invoice received in mail
4-6-78 Commodore shipped one PET
4-12-78 UPS delivered to Ft. Worth home PET 2001-8 SN0013739

Elpased time = 22 days

Personal Computing magazine order vs. delivery

3-25-78 $14 check mailed to *PC* for 1 year subscription
5-11-78 May issue – 1st magazine of subscription received

Elapsed time = 47 days

Moral: "Software takes twice as long as hardware."

John R. Massie
Forth Worth, TX

Editor's note: Your check arrived after subscriptions for the April issue were sent to the circulation department's computer. Your subscription was therefore entered beginning with the May issue. We require a maximum of two months for processing, as do most monthly magazines, so we're happy when a new subscriber receives his first issue in just 47 days.

Music Encore

Dear Editor:

My thanks to Linda M. Schreiber for her music transposer article (May PC). I can put her program into my FSMUSIC which plays tunes from the DATA statements. I have a BASIC compiler written for the PET that sets up the output interface to generate tones of crystal-controlled precision with middle A at exactly 400 Hz. It will only go down to B below middle C, but up off the piano keyboard. My data statements look like "400 DATA C#, 2, B, 2, C#, 2" where the number is the length of the note – 1 for 1/8 note, 2 for 1/4, 3 for dotted 1/4, and so forth. My present compiler goes from BØ to G1# where BØ is below middle C and G1 is above middle G. All accidentals are written as sharps.

Frank Alexander
Swarthmore, PA

Talking PET

Dear Editors:

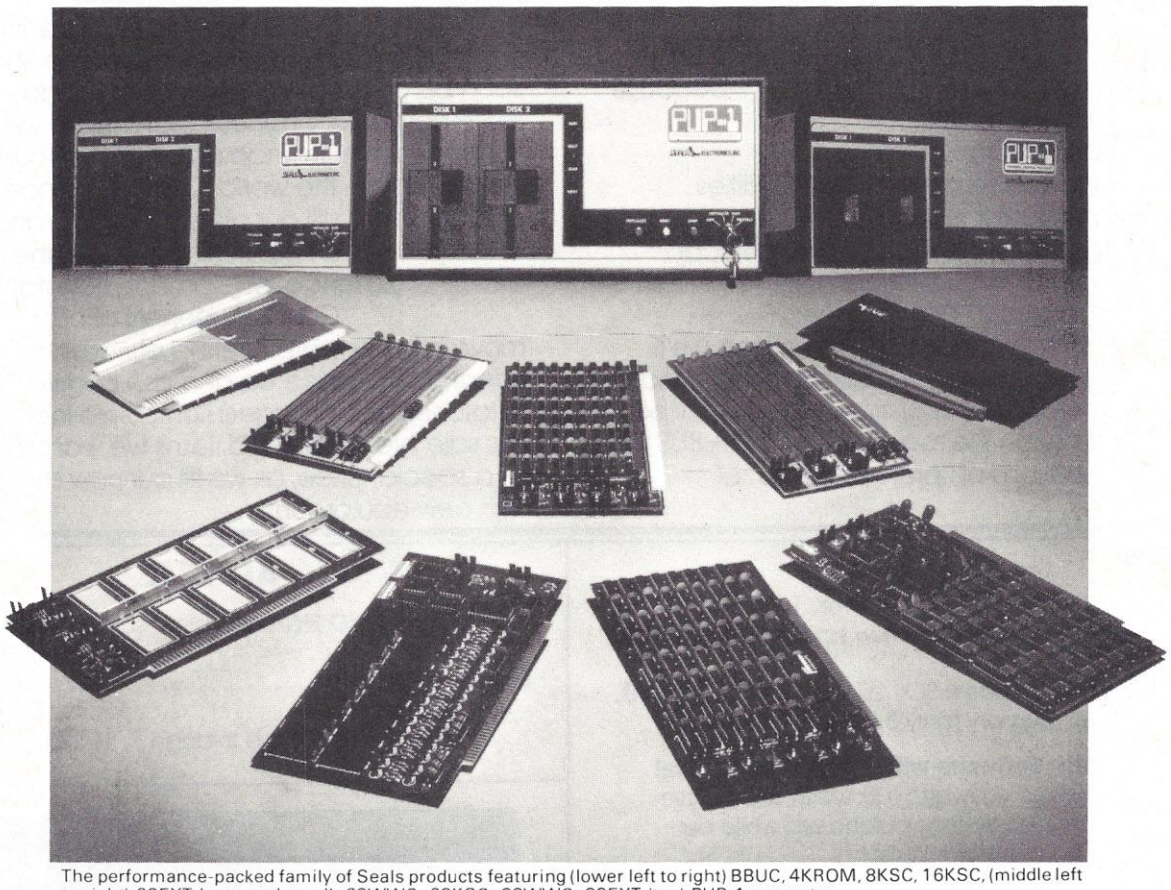
We have finally made our PET speak (in print) using a Diablo daisy wheel printer interfaced with Dick Rosner's PET ADA from Connecticut Micro Computer, Brookfield, CT.

The device converts IEEE 488 output to RS232 and works very well. Our initial start up didn't work because two pins on the Diablo (6 & 4) must be jumped. We also had trouble with tabbing, line feed, form feeds, etc. Diablo uses ESC and a character to perform these functions. To accomplish tabbing, use Print #5, CHR$(27); CHR$(09); CHR$(B) where B is the variable position on the page and CHR$(27) is ESC in decimal form and CHR$(09) is HT on horizontal tab. This sequence causes an absolute tab to the horizontal position on the line. By using both absolute vertical and horizontal tabbing, the print position can be quickly located anywhere on the page. The key to the typewriter control is to use CHR$ and decimal equivalent of the keytop sequence. Similar statements cause the Diablo to print in Red, Black, Sound Bell, Carriage Return, etc. We scratched our heads 'til a call to Connecticut Micro Computer put us on track.

Although the PET ADA may be a bit expensive at $169 for a complete unit including case and power supply (stripped for $98), it is our first usable short cut to print on a quality printer.

Bueck/Jenkins
Rocky Mount, NC

What you should know about the Seals Retail Program.



The performance-packed family of Seals products featuring (lower left to right) BBUC, 4KROM, 8KSC, 16KSC, (middle left to right) 68EXT large and small, 68WWC, 68KSC, 88WWC, 88EXT (top) PUP-1 computer.

At Seals Electronics we believe in specialization. We concentrate on manufacturing a reliable line of microcomputer products so that you, the retailer, can concentrate on sales.

When you are a Seals dealer you can depend on Seals for:

- Dependable, quality hardware with an industry wide reputation for excellent performance.
- A marketing program designed to support you at the point-of-sale.
- Price margins that give you the best possible advantage and recognize that you need profits to operate.
- A staff of marketing and technical specialists who are interested in you and your business.

The people at Seals have been in the microcomputer business for a long time. We know that the computer retailer needs and deserves the very best possible support from a manufacturer. We are constantly on the alert to improve our retail program. Seals works with you to increase your sales and assure you satisfied customers. We don't ask that you be a manufacturer (all our products are available assembled and tested) and we know you don't want us to be a retailer.

The 1977 Computer Store Survey published by Image Resources gave Seals Electronics consistently high ratings in the areas of product image, value to customers (product reliability and documentation), and dealer interface with manufacturers.

We are proud of our record with retailers and are working hard to improve our position in the industry. We would like to work with you.

For current literature on the Seals microcomputer product line and/or more information on our retail program, call or write our Marketing Department, Seals Electronics, 10728 Dutchtown Road, Concord, TN 37922, (615) 966-8771.

Dealer support is more than just words to us.

SEALS ELECTRONICS
INCORPORATED

CIRCLE 10

We've Undergone a "Moving Experience"

South Carolina's Byte Shop has moved to a new location. We have a new store, with three times the floor space.

Our "moving experience" was undertaken with our customers in mind—because now we'll be able to offer:

- a large new showroom
- expanded service staff and facilities
- new product lines
- a large "browsing library" of technical periodicals and publications.
- UPS shipping, 5 days a week

But with these new services, we'll continue to offer our same old personal attention to every order, regardless of size. And we'll still be putting our show on the road with our DATA BUS for field demonstrations of "preferred lines" of equipment.

In addition to our move, another signal event is that the Columbia Byte Shop will soon be celebrating its second birthday. In the small computer business, that means we've been in business longer than about 80% of our competitors in the world. Our experience in sales and service is still our greatest asset—and yours, when you need help sorting through the myriad claims of dozens of equipment manufacturers.

One final benefit of our recent move is that we are offering our customers a **special sale price on** demos, used items, and scratched or damaged items. Also included in this sale are several old items we want to "move" at a special price, as we fill our new showroom with new equipment.

SOL 20's

Demo System - including SOL 20, 16K, T.V., and cassette—works fine but has a few scratches . . **$1995.**

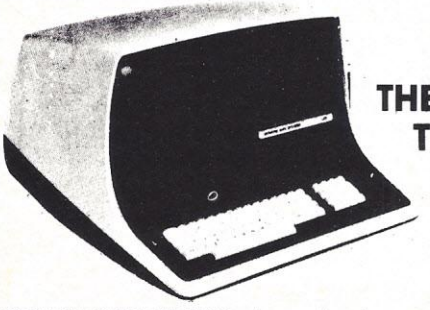
SOL 20 Kits at OLD Prices: SOL 20/16K . . . . . **$1550.**
or $1150. without memory (only 3 left).

Free PT Cassette Software with system purchase!
That's right, we will give away $100. worth of software with each assembled system purchased while our cassette software overstock lasts.

APPLE II

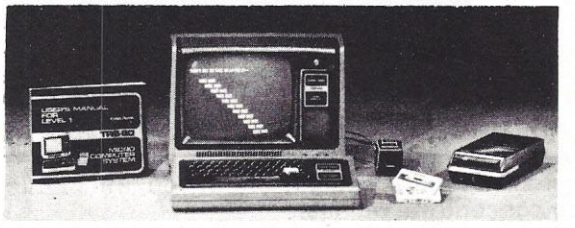
16K APPLE DEMO SYSTEM (minor scratches) . . **$1050.**

New 16K Apple II . . . . . **$1195.**
Apple Disk . . . . . $495.
(guaranteed price through 7/31/78)



THE INTERTUBE TERMINAL

$784.



Now that you've gotten a glimpse of what a computer can do — trade in your TRS 80 or PET on a **Cromemco, Poly 88,** or **SOL 20.** Move up to Fortran and Cobol. For a limited time we will allow the full purchase price of a basic TRS 80 or Pet when purchasing a complete Cromemco, Poly 88, or SOL 20 system.

GREEN CRT FILTERS - low reflectance surface with tint to produce the appearance of "Green Phosphor" are available for 9 and 12 inch monitors **$14.95**

SUPER SYSTEM SPECIAL

$1495.

Apple II (16K), G.E. Color T.V. (10" diagonal), G.E. Cassette Recorder, and M.&R. Super Mod II. This is a complete color graphic system — plug it in and RUN.

POLY 88's

—1 System 16 Demo (some scratches) including Hitachi T.V. with Pickies and Trout Mod, and Cassette . . . . . **$1650.**

—1 System 8813 one drive (walnut cabinet damaged and older style keyboard . . . . . **$2499.**

—4 System 16's - new but have older style keyboards . . **$1850.**
—4 8813 Upgrades @ ($1250 if purchased with the above system 16's) . . . **$1450.**
—5 VTI 64 Video Cards . . . . . **$195.**

CALL 1-(803)-771-7824 TO ORDER OR WRITE

BYTE SHOP™ #32

master charge

VISA

1920 Blossom St. Columbia, S. C. 29205

CIRCLE 11

RANDOM ACCESS

PC on the air

A television first — a casting call for robots! *Personal Computing* magazine is going on the tube and to host its new television series on how to use small computers in the home for fun and profit, the magazine is auditioning mechanical men and women.

Program producer Norm Blumenthal, who has tentatively set the show title as "Computer Base", said, "There are about 2500 robots presently engaged in productive or entertainment jobs around the country, and they are all eligible to audition."

According to Blumenthal, selection will be made on the basis of a variety of factors with points awarded for such traits as intelligence, attractiveness, personality, poise, tonal quality of voice and quickness of wit. No insensitive robot-like robots need apply.

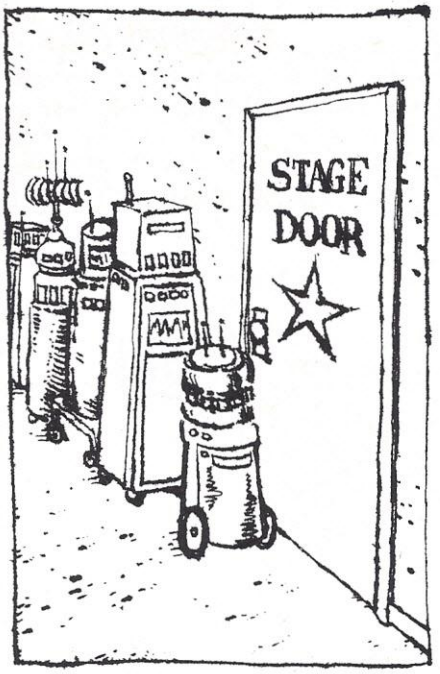
The program is scheduled to go on the air with nationwide syndication in the Fall. A noted personality will co-host with the robot.

"Computer Base" will show viewers how to get the most out of their home and business computers, combining the elements of a variety show, TV game show, instruction seminar and network sports show as it demonstrates the capabilities and applications of small business and home computers. Articles appearing in *PC* will be adapted to television format for the show.

Blumenthal is a former executive producer at NBC, who for fifteen years had total responsibility for "Concentration", television's longest continually running daily game show. Blumenthal said, "We will teach the audience how to make money, how to make merry and how to make music."

Each program will have a segment devoted to "The Lemonade Stand", illustrating how enterprising individuals from pre-teenagers to senior citizens can use their microcomputers to conduct small businesses.

"Viewers will also be informed on how to program computers to compose songs, poetry, short stories and even novels. They will learn how to produce effortless gourmet meals and how to use a computer to perform hundreds of otherwise time-consuming household chores," Blumenthal said.



Robot host and guest personalities will vie with viewers in a variety of game contests on both an individual and team basis. Audiences may climb in the electronic ring against a computerized Muhammad Ali, serve aces in the center court at Forest Hills against a Jimmy Conners or even match quarterbacking wits against a Roger Staubach in a mythical television "Futurebowl".

"In addition to providing excitement, entertainment and education, we will help eliminate much of the fear that the general public has about the mechanistic, impersonal character of computers by demonstrating that, despite their wondrous microsecond performances, computers can only respond to the manipulative power of man. What goes in is still what comes out," observed George Palken, President of Benwill Publishing Company, which publishes *Personal Computing.*

If you have a robot you'd like to audition for the co-host spot, contact Bob Berman, Personal Computing, 1050 Commonwealth Ave., Boston, MA 02215, or call (617) 965-2608 or (617) 965-4400.

Hit me – for a fee

A new coin-operated electronic game computerized to play blackjack has been introduced by Computer Kinetics of Westlake Village, California. Called Vega 21, this non-video machine contains a microprocessor programmed to count, calculate, duplicate real odds and monitor the progress of the game via a display screen.

Vega 21 acts as dealer and scorekeeper. An instruction plate shows how many points credit and how many hands are allotted for each coin inserted. The game continues until the player runs out of credit points or has used up the allotted hands.

The machine allows splitting of pairs, doubling down and insures dealer aces.

It is programmed to test itself to make sure it is functioning properly. Any problem will show up on the message screen. There is a single door on the back panel that provides access to the coin vault. Servicing of any internal machine part can be accomplished from the top. Overall size of the unit is approximately 17 inches deep, 17 inches wide and 11 inches high. It weighs 20 pounds.

Illustration by Casserine Toussaint

Noise pollution solution

A jumbo jet loaded with passengers bound for New York speeds down the runway at Honolulu International Airport, engines at full thrust. At the same time, sensitive microphones near the runway begin monitoring the engine noise, and a computer inside the airport control tower starts recording the decibel levels at half-second intervals.

As the plane gains altitude, microphones — at the state capitol, Diamond Head crater, and 15 other locations — continue to monitor the plane's noise "footprint".

In this particular instance, all locations register no excessive noise from the big jet.

Phone lines link the remote microphones with the airport computer. Two lines extend from each microphone, (atop 20-foot-high wooden utility poles) one is for continuous data transmission, the other for checking noise sources and acting as a back-up.

Honolulu's aircraft-noise analysis program, a $200,000 project underwritten by the Hawaii State Department of Transportation also compiles information on aircraft flight patterns, wind conditions and time — all the data necessary for complete analysis of factors that produce excessive noise.

While the noise abatement system captures all kinds of sounds, from barking dogs to raucous Mynah birds, the computer is programmed to ignore these unrelated sounds and only pick up those from aircraft.

In addition, daily average noise levels at each site are calculated and periodically printed by the computer, so long term trends can be identified and noise reduction goals verified.

Unlike other airports where airlines are fined for causing excessive noise, Hawaii's system is not intended for regulatory use. Airport officials regulate flight patterns because Hawaiian citizens are far more vulnerable to aircraft noise than their mainland counterparts due to the unique year-round open window climate. However, the state does provide the airlines with data on specific flights that exceed acceptable noise levels.

So you can let your Mynah squawk all it wants, but keep your Lear jet down to a dull roar.

I'll put you on hold

A telephone with built-in microprocessor and an eight-digit numeric display has been introduced by Rolm Corporation.

Called the ETS 100, the unit can replace multiple-key business telephones and costs about the same as many conventional key telephones. It is designed to be used with the ROLM CBX computerized business telephone system.

ETS 100 reduces costs by displaying elapsed time on toll phone calls and speeding up function changes on extensions. Messages can be left electronically on another person's phone — for example, "call 1234".

New lines can be added to an existing unit, extension numbers changed and functions modified by typing instructions on a computer terminal connected to a ROLM CBX, with no change in hardware required.

Because both the unit and the system are computer-based, new features and down-stream capabilities can be added without requiring new hardware. In addition, function buttons on each instrument can be customized, extension numbers switched around and telephones restricted to internal or local calls only, all by a service person using the maintenance teleprinter.

Other features, include "group pick up", allowing secretaries to answer any ringing extension, an automatic connection to the ringing line by lifting the receiver, a "voice call" feature permitting the basic ETS to function as an intercom, and "speed dialing" — calling commonly used numbers by de-

pressing two or three buttons.

The eight-digit LED display indicates "call me" messages, the extension number which is calling you, a call is waiting, the call was forwarded from a busy extension, the phone has been put on "do not disturb", the number dialed, time since call began, and the time of day.

Account codes can be entered without interfering with a conversation because the ETS uses direct digital instead of the conventional analog tone for dialing.

ETS contains 2000 words of programmable read-only memory and 256 words of random access memory. It measures 9-1/4 inches wide, 9-1/2 inches deep and 5-1/2 inches high and has 32 buttons — 12 for key-pad (dialing), 16 for features, 3 lines, and a hold.

Corporation officials say the instrument is the precursor of a desk telephone which will perform such functions as data entry and retrieval, interfacing to work processing devices, electronic mail transmission and even credit card verification.

Additional information can be obtained from Rolm Corporation, 4900 Old Ironsides Drive, Santa Clara, California 95050.

Orient Express

Want to know what's happening, computerwise, in Singapore, Hong Kong or China? Try the *Asian Computer Yearbook.*

This one-volume, 200-page reference contains country-by-country data on the state of computer arts in Asia. Over 1000 directory listings cover Singapore, Malaysia, Philippines, Thailand, Indonesia and Hong Kong. A special report details China's home computer industry, while another report gives technical information on a computerized betting system at a Hong Kong horse racing track.

For more information, contact *Asian Computer Yearbook,* Computer Publications Ltd., Seabird House, 7th Floor, 22 Wyndham Street, Central, Hong Kong.

Computers and You

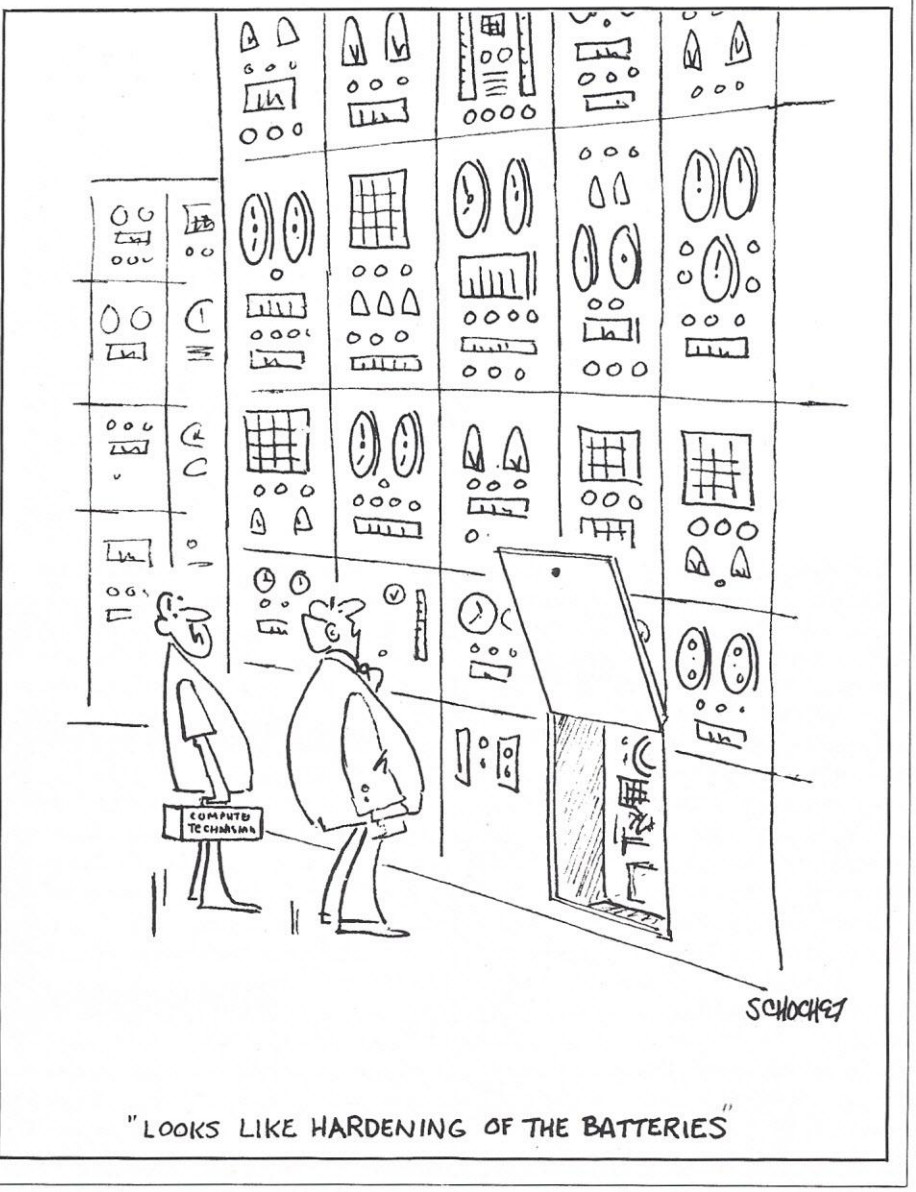
To help take some of the mystery out of computers, Canada Systems Group developed a course called "Computers and You". Designed for non-computerists whose work brings them in contact with computers, the course teaches basic knowledge of computers and data processing.

In one and a half days, the course provides an overview of data processing so that students can work more effectively with computer people. The roles of professional computerists such as operators, programmers and analysts are discussed, with emphasis on the importance of the close interface between users and data processing personnel.

Students learn terminology and come into close contact with actual hardware, storage media and working documents used by computer people. Other course segments discuss common business systems — retail sales, credit card processing, check processing and bank statement production.

According to Canada Systems Group, the course "creates a good business and personal attitude towards computers to help students perform more comfortably and productively in a data processing environment."

For more information, contact Registrar, Canada Systems Group Education Centre, 2599 Speakman Drive, Mississauga, Ontario L5K 1B1, (416) 822-5200.

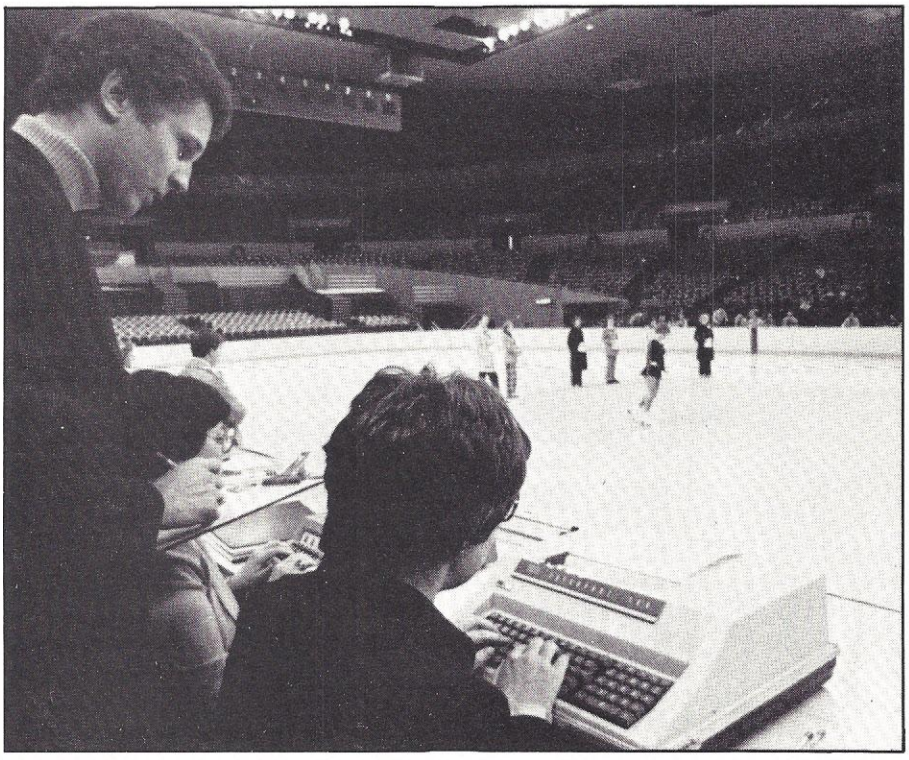


"LOOKS LIKE HARDENING OF THE BATTERIES"

Computer wins Skating Fans

A competitor's split-second timing can make the difference between winning or losing a figure skating event. Similarly, timing in the scoring process can make the difference between winning or losing audience interest in figure skating as a sport.

Now, thanks to A.L. Beard of Honeywell, scoring can be programmed, which speeds up the event considerably. Before computerization of the scoring process, it took up to an hour to calculate championship standings. The public tended to lose interest during the interim.



Since 1972, when the U.S. Figure Skating Association began testing the computerized scoring system in parallel with manual methods, Honeywell has donated computer time to run the programs.

Scoring of figure skating events is more complex than the high/low discard and average system of gymnastics and diving. Each event is decided by a panel of judges, typically seven or nine. Each judge evaluates the skater independently and grades the performance on a scale of 0 to 6 in one-tenth increments. The highest mark goes to the best skater in the event.

In free skating and free dance events, two marks are awarded to each skater or couple from each judge, one for technical merit (difficulty, variety, cleaness and sureness), the other for composition and style (harmonious composition, conformity to the music utilization of space, originality). Compulsory figures and most compulsory dances are scored with only one mark.

The judging system is supported by a scoring system carefully designed to find a consensus among the judges and minimize the effects of judges inconsistent with the majority.

The computational part of the scoring process is handled by a group of skating association volunteers who are designated accountants. Two of them operate the two rink-side terminals that connect by a telephone line to two Honeywell computers on a time-sharing basis.

Results are printed on the terminals about 30 seconds after all the data is entered. Results must then be certified by the chief accountant and referee, signed, photo-copied and distributed. The entire process takes 5 to 10 minutes.

Final results are most important, but the value of a computerized scoring system is most apparent in the posting of intermediate standings, something not available in pre-computer days.

According to Beard, the spirit of competition among the skaters is increased by the availability of intermediate standings.

Before computerization, no official standings were available throughout the 3-1/2 hours it took to complete a round. The judges' marks were available for each figure, but marks are not standings. The skaters and the public wanted standings.

The rink-side terminal operators are familiar with skating but not with computers. The programs prompt the novice operator in skating terminology and make all input and output parallel the previous manual formats as much as possible.

Processing for each championship event involves event creation (entering names, numbers of judges, event type, etc.), multiple instances of marks entry and results computation, and event wrap-up (final report, generation of specially-computed standings, etc.).

The programs are written in FORTRAN. The operator directs the program by entering an event code (or file name) followed by a numeric command to the program.

An operator's guide sheet lists all commands, a description of what they do and a two-digit number for invoking a function. Commands are designated "functions" and their two-digit representations are called "function numbers."

Operators quickly memorize the more important function numbers. The net effect of the design is that skating association people who are familiar with the scoring process but not with computers become terminal operators after a few hours. They learn how to direct the programs and get results even though the facilities they are directing are capable of many complex functions, Beard said.

Computer Myths

Not all text books fall within the field of "stand-alone" books. Usually, the text is tied intrinsically into the guided discussion of a teacher. One text-book that can stand alone, and can be read at leisure without benefit of a lecturer standing nearby, is Prof. Patrick Henry Winston's *Artificial Intelligence.* © 1977 by Addison-Wesley Publishing Co. Reading, MA. Director of M.I.T.'s Artificial Intelligence Laboratories, Dr. Winston can be expected to treat the fascinating subject of artificial intelligence with expert scrutiny. He succeeds in making the subject one that intrigues anyone with any amount of interest in this field. Can a computer stand on its head on its own volition? Well, Dr. Winston points out that the machine is capable of doing many amazing things, though standing on its head is not mentioned. What he does mention are entrancing subjects such as "Computers Can Learn", "Computers Understand Simple English", and "The Simulated Psychiatrist". He closes the excellent treatise with a dialogue between the "proceduralists" (knowledge about how to do things by programming is procedural) and the "declarativists" (who believe in storing facts in big data bases from which programs can be generated).

DECLARATIVIST: "No one taught you that to divide and understand is the key principle of science. I'll suffer a bit with my way of keeping knowledge about processes because my way allows extensibility . . ."

PROCEDURALIST: "Bah! You're right about dividing and understanding. Nearly decomposable systems are the stuff the world is made of and I agree that the best way to approach most problems is by studying the pieces independently . . ."

There are many interesting and attention-holding sections of Professor Winston's book. One such section is called "Myths about thinking":

Myth: Computers Can Never . . .

It hardly matters how the sentence is finished because the standard proof is as weak as it is inevitable. Stripped of obfuscation, it goes like this: "Computers cannot . . ., because no one has thought of a way to make them . . ." The elusive qualities most frequently proposed by the critics of computer intelligence include learning, introspection, and aesthetic feeling, all of which suggest a certain unfamiliarity with the literature . . .

Of course to believe in human superiority is a tradition. Once our earth was the center of the universe, now it is an undistinguished planet. Once our creation was direct and divine, now some people believe it is the good luck of the primates. Once our intelligence was unchallenged, yet someday computers may laugh at us and wonder if biological information processors could be really smart. Beware of those who think it can never happen. Their ancestors hassled Galileo and ridiculed Darwin.

Myth: Computers Are Not Intelligent Because They Do not Write like Shakespeare, Compose like Beethoven, or Do Science like Newton

Logic dictates that true demonstration of computer intelligence need not hinge on producing superhuman computer performance. Otherwise we ordinary writers, musicians, and scientists would necessarily and painfully find ourselves among the unintelligent.

Myth: Computers Can Do Only What They Are Programmed to Do

Intelligent computers do not organize themselves out of nothing, so in some uninteresting sense their abilities descend from human programmers. But it is equally true that humans are indebted to the genetic code. Somehow there must be enough innate information processing power to get beyond the threshhold above which learning from the environment takes place. Once humans bring computer intelligence up to this level, computers will no doubt augment their directly programmed gifts by the same means humans do: by being told, by reading, by asking questions, by doing experiments, and by being curious.

Myth: Software Can Never Equal Brainware Because Transistors Are Different from Neurons

. . .There is an argument against the idea that studying neurons can lead to much of an understanding about intelligence. Understanding them beautifully and entirely can no more produce an understanding of intelligence than a complete understanding of transistors can yield insight into how a computer can understand scenes or respond to English. People cannot think if we pluck the neurons out of their brains; but if we study only neurons, we have only a slender chance of getting at intelligence.

Still, some critics argue that computers cannot be intelligent because digital hardware made of silicon never do what brains made of neurons do. Their position is weakened by the hierarchy argument and the lack of solid knowledge about what the unthinkably tangled neuropil does.

Continued on next page

Illustration by Casserine Toussaint

Myth: Probabilistic Machinery Causes Inspiration and Explains Free Will

People love explanations. Even when something defies analysis, default explanations fill the void, allowing uncomfortable puzzles to be settled with a minimum of fuss. Many ancient Greeks supported Socrates' opinion that deep, inexplicable thoughts come from the gods. Today's equivalent to those gods is the erratic, even probabilistic neuron. It is more likely that increased randomness of neural behavior is the problem of the epileptic and the drunk, not the advantage of the brilliant. Powerful computer intelligence will rest on description, representation, problem solving, and other powerful ideas, not on disorder.

Myth: Computers Can Never Appreciate Aesthetics

It is argued that creative arts of various sorts never can be approached scientifically. Beauty must be felt, it cannot be reduced to rules and rational thinking. Even the attempt eviscerates.

Certainly some artists find it difficult to vocalize what they do, but this is not a proof that computers can never create or enjoy a painting, a play, or a symphony. Indeed smart computers will undoubtedly find art a challenge since descriptions and interactions of descriptions must surely be central to understanding why any art form is interesting, moving, pleasing, disquieting, or new.

In any event, if computers do get good at artistic craftsmanship, there may be some question about wherein the art lies. If a person writes a program capable of 10,000 string quartets a day, do those quartets constitute the art? Probably not. Oddly, but inevitably, the art will be in the program.

Myth: Intelligence Can Never Be Understood

Of all the myths, this is perhaps the nearest to truth, but for an unexpected reason. To be intelligent is to be mysterious. "How did he know that?" we say. As long as the origin of an idea is obscure, its invention seems profound, but as soon as the explanation surfaces, we wonder, "Why didn't I think of that, it's trivial!" As soon as a process is dissected, studied, and grasped, the intelligence invariably seems to vanish.

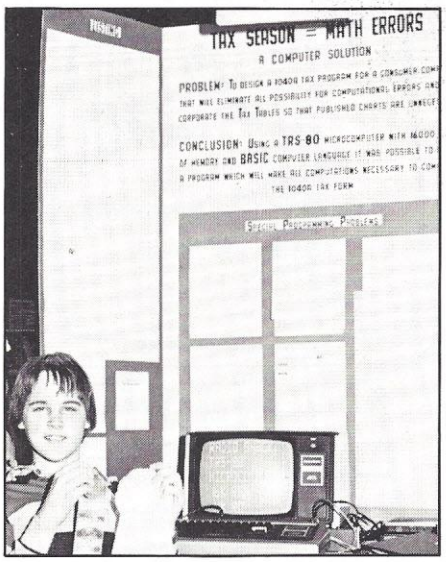
Much the same happens when programs are studied. Vintage performance becomes *vin ordinaire* once details are exposed and limitations are seen. Instead of embracing a system's intelligence, study dilutes it. One must recognize this natural tendency

— Harry Shershow

Computer Prodigy

It took a 14-year-old to figure out a painless way to complete your tax forms. Eighth grader Ralph Lipe of McLean Middle School in Fort Worth, Texas, received top honors in a recent Science Fair competition with a computer program for the 1040-A tax form.

Using a Radio Shack TRS-80 Microcomputer System, Ralph wrote



a program designed to guide a person step by step through the 1040-A tax form. It took him a total of 60 hours to write and de-bug the program. The idea came from his mother, a certified public accountant, who assisted him in setting up some of the equations for the tax questions. Ralph then converted the information to BASIC language for programming the TRS-80 microcomputer.

Ralph's project captured the first-place ribbon in the Math and Computers category and was chosen "Best of Fair" out of approximately 200 entries in McLean Middle School's intramural science fair. At the Fort Worth Regional Science Fair, which hosted 460 entries from seventeen counties in North Texas, Ralph won a first place ribbon in the Math and Computers category, a first place ribbon from the Institute of Electrical Electronics Engineers and placed "Third Best of Show" for the Junior High School division.

Ralph, who someday hopes to run a computer for NASA, thinks computing is "almost like a sport that everyone can enjoy". His own interest in computers developed last year when his father brought home a microcomputer.

According to Ralph's father, when Ralph first sat down at the microcomputer, he knew nothing about making it operate. Within a few hours he'd taught himself to write a few simple programs using the instruction manual furnished with the system.

TRS-80 Notes

Just buy a TRS-80 and need some help with your programming? Or are you an experienced user who needs some advice or has some to offer?

In either case, you may be interested in a nonprofit, no-advertising users' newsletter published for TRS-80 users.

The newsletter, operating independent of Radio Shack, will provide a medium for people to express opinions or offer ideas on programs, accessories and other publications.

The editors also publish a word processing newsletter, as well as user notes for S-100 and S-50 buses. The user notes are $10 US ($18 US overseas) for 12 issues; Word Processing is $12.95 US ($20 US overseas) for 12 issues; S-100 and S-50 publications $5 ($10 overseas) for 6 issues. Checks should be payable to Bookmakers, Box 158, San Luis Rey, CA 92068.

Robot Turtle

R2D2 it's not, but the Turtle, a small home robot invented by MIT Senior William D. Hillis, may be the next best thing. The Terrapin Turtle, a commercial home robot ready for sale, can walk, talk, blink, feel and draw, according to Hillis, who is affiliated with MIT's Artificial Intelligence Laboratory which has used these robots in various projects. After producing a commercial version of the Turtle, he joined with David L. McClees, a senior at Harvard, to form Terrapin, Inc., to manufacture and sell Turtles to home computer hobbyists.

The Turtle consists of a chassis topped by a hemispherical dome 3-1/2 inches in radius. It has two large wheels and can move six inches per second. By turning the wheels in opposite directions, the Turtle can be made to rotate in one spot. Other features include beeping its speaker and flashing its lights.

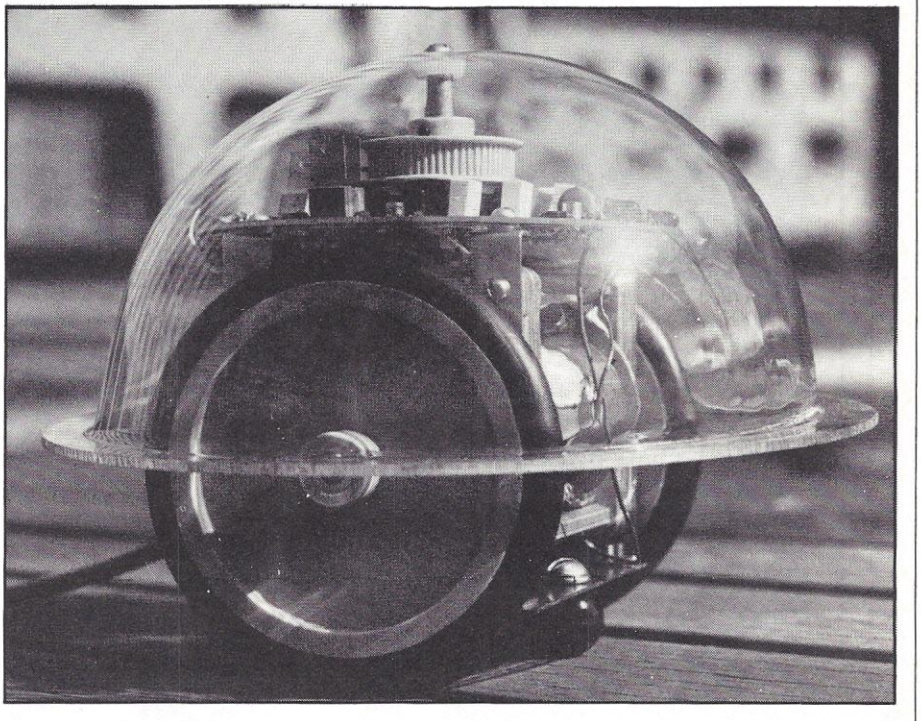
The robot can "feel" by using its dome/shell as a touch sensor so that it "knows" when it bumps into an object. To draw, it lowers a pen from its chassis and moves. With these two capabilities, the Turtle can map rooms by entering them, bumping into the objects within them, and storing the configuration in computer memory. Then the Turtle can let down its pen and draw a rough representation of the floor plans.

There's one catch, however. The Turtle needs a computer to provide directions. Both Hillis and McClees see the Turtle as a valuable addition to any home computer buff's hardware. A kit will retail for $300.

While the officers of Terrapin, Inc., think that Turtles will be popular on the home computer market, they hope that the educational uses of the Turtle will not be overlooked. The Turtle can be used to teach simple geometry and computer programming concepts, they said.

Hillis and McClees are hoping that the Turtle will help make computers seem like more than just nasty machines that make mistakes on bills which is how some people view them.

High school classmates, Hillis and McClees said they wish Turtles had been around when they were learning about computers.



Tracing students with GEMS

Over 7,000 grade-school students in Sandy, Utah, are now monitored by a Goal-based Educational Management System for reading, math, English and science. The computer program, TRACER, helps trace student progress through the goals and objectives of the programs.

The program provides a research base for principals, teachers and curriculum staff to examine the objectives of the GEMS program, the technical soundness of test items and instruction effectiveness.

TRACER is based on the principles of mastery learning, which assumes that 80% of the students can learn what formerly only 20% were expected to learn. It helps teachers provide different instructional strategies and learning times for each student. Independent learners can plan their own programs of study.

Using special computer cards, students and teachers report information about student status, record answers to test questions and indicate any changes in a student's program. The teacher receives a printout with information about the test results, grouping, conferencing needs and any comprehensive progress reports requested. Teachers estimated that for a class of 25 students it would take over five hours to duplicate the record keeping done by the system in less than 5 seconds.

TRACER does not teach, but is a classroom aide which frees teachers from clerical work and allows them to spend more time with students. It gives teachers technological support without reducing their professional status. Most GEMS units have criterion-referenced tests which are scored by the program. However, the teacher can override the system at any time and make a subjective evaluation on a student's mastery of units. For units without referenced tests, the teacher determines when mastery is achieved.

TRACER is distributed by CTB/McGraw-Hill, Monterey, CA. It is also used in Anchorage, AK, Riverside, CA, and Pasadena, CA.

Heavenly HOST?

An electronic HOST is taking care of reservations for the Biltmore and Roosevelt hotels in New York City. The hotels — which together contain more than 1900 rooms — are using an automated hotel management system and two Centronics 306 printers to facilitate day-to-day operations and to preregister guests who make their room reservations by telephone.

HOST — Hotel On-Line System Technique — consists of a Century 201 computer and more than 50 CRT terminals. The printers enable hotel employees to perform other functions more vital to the Hotel's guests.

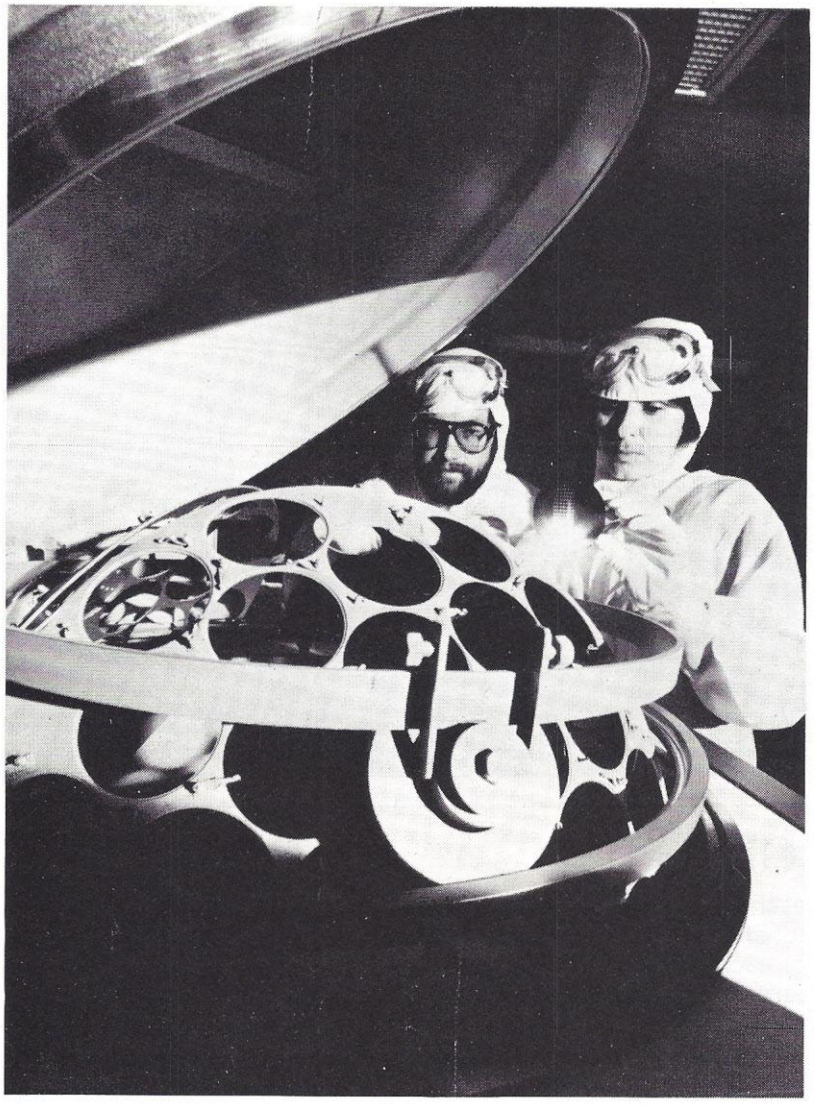
The system becomes effective even before a prospective guest completes his telephone reservation request. As soon as the desired arrival date and time are known, one of the seven reservation takers accesses the system to determine if a room is available on that date. The answer appears on a CRT screen located at the work station, which enables the employee either to confirm the reservation or, if a room is available only at the sister hotel, to recommend that the caller stay there.

Nearly 2,000 rooms are available to guests in two separate hotels on any given date, resulting in a high percentage of caller accommodation.

On the morning before the guest's arrival, the printer generates the registration card, containing name, address, type of room, rate and special instructions, if any. This printout appears on a three-part form, with copies going to the bell-hop, accounts receivable and the front desk with the guest's folio. Throughout the guest's stay at the hotel, the computer monitors every expense — from bar bills to telephone calls to restaurant charges. At the end of the stay, a bill is computed by printer when the individual checks out.

Another feature of the system is its recall capability. If a registration card is lost or misplaced, the computer simply instructs the printer to recall that specific card, and it's produced again.

So if you're planning a trip to New York, consider staying with a congenial, if electronic, HOST.



Space-age technology enables engineers to design faster, smaller computer circuits. This machine deposits a layer of conductive metal only 40 millionths of an inch thick on a wafer of silicon, just one step in circuit fabrication. Burroughs' newest B 6800 computers use memory circuits no bigger than an aspirin tablet. Yet, each circuit, or "chip", contains a hundred times more components than a television set, provides over 16,000 microscopic positions for storing data and allows data to be read by the computer in only 300 billionths of a second.

1802 User's Group

A COSMAC-1802 user's group is forming for all 1802 users, including BASIC ELF, ELF-2, Super-ELF, VIP and UC-1800. Membership is free. The group plans to provide correspondence, an exchange of ideas and software and possibly the publication of a news newsletter. Write to Patrick Kelly, P.O. Box 7162, Los Angeles, CA 90022 for more information.

CRT/TERMINAL

Fully Assembled And Tested ! ! !

$399.00

Full 90 Day Warranty

STANDARD FEATURES:

- FORMAT DISPLAY 32 X 16 SWITCHABLE TO 64 X 16
- SCROLLING AND PAGING
- 6802 MICROPROCESSOR CONTROLLER
- RS232 I/O PORT
- ALL POPULAR CRYSTAL STABILIZED BAUD RATES
- UPPER AND LOWER CASE
- 32 ASCII CONTROL CHARACTER DECODING, OUTPUTS AVAILABLE TO USER

OPTIONS:

- NUMERIC KEY PAD INCLUDING LETTERS A TO F $55.
- MODEM-ACOUSTIC OR 4 WIRE $100.
- 5 SLOT S100 BUS BACKPLANE INCLUDING HIGH CAPACITY POWER SUPPLY $155.
- 16 COLOR DISPLAY $45.
- CURRENT LOOP $10.
- GREEK SYMBOLS $25.
- 9" OR 15" SANYO B/W MONITOR IN METAL CASE

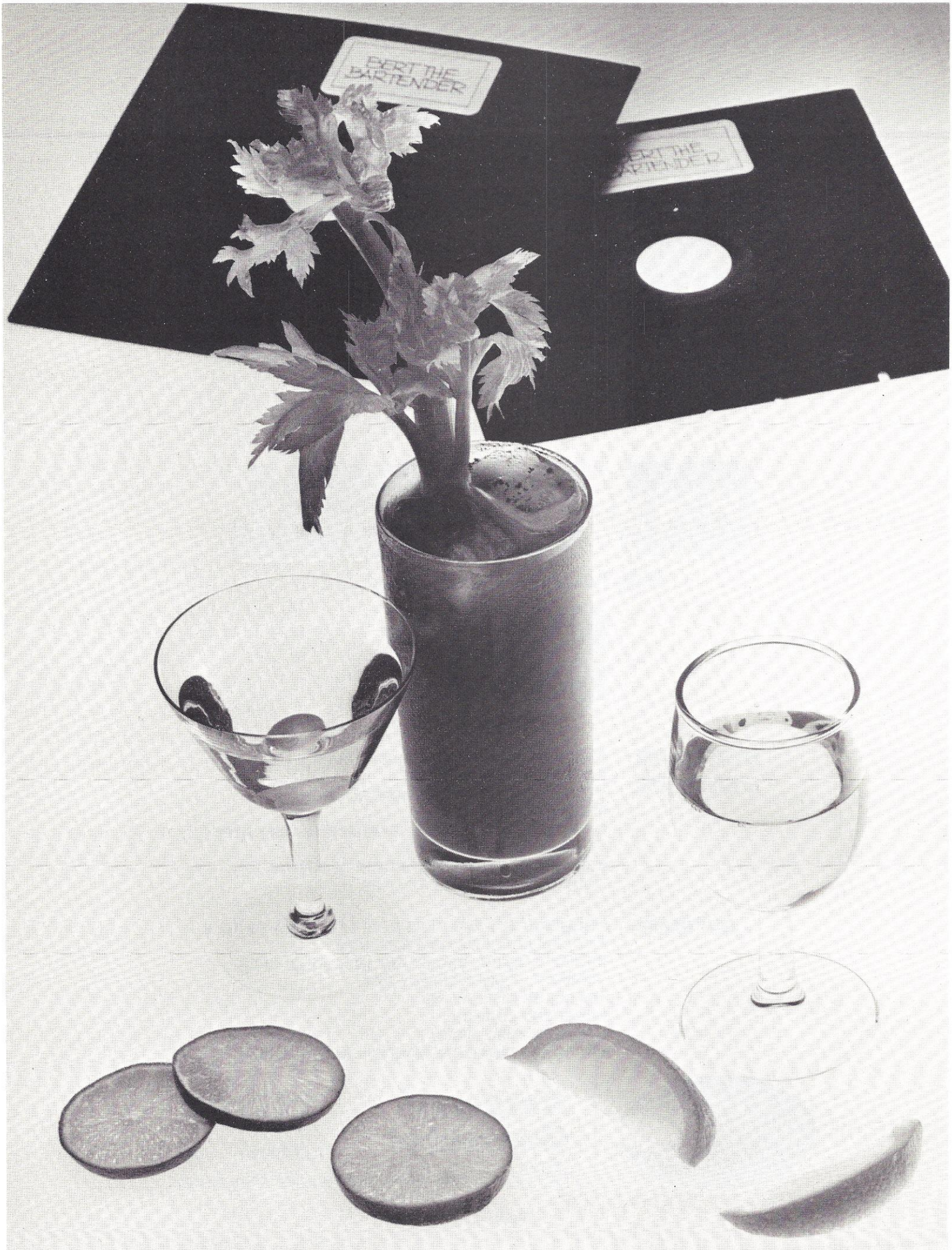
Caldwell Computer Co.

546 West Olney Avenue
Philadelphia, PA 19120
(215) 224-9133

Sam Caldwell

SEE US AT THE PHILADELPHIA SHOW

CIRCLE 12


BEST THE BARTENDER

COMPUTING ON THE ROCKS

BY SAM NEWHOUSE

Whether you are planning a party for 100, a get-together for six, or an evening at home alone, this program will help you prepare your before-, during- and after-dinner drinks with the expertise and flourish of a bartender with 20 years experience.

Instructions for mixing everything from a Gold Cadillac to Russian Coffee as well as martinis and gimlets are at your fingertips.

You can call up a drink by name, or just input the main ingredient you like to use and the program will list the available choices.

"Barkeep" supplies the recipes for your chosen drink along with the garnishes that should accompany it and what type bar glass you should serve the drink in. For the novice not familiar with the various bar glasses, the program will even print, on command, a picture of each.

Finally, the program will print you a shopping list for the quantity of ingredients necessary to prepare the number of drinks you specify. All amounts are listed in ounces unless otherwise noted.

The program ends with a listing of common bar measurements – in case you forget a pony equals 1.5 ounces and a jigger equals one ounce – or is it the other way around? Read on and find out.

Using the Program

Specify any drink name the program knows, and its recipe will be displayed. Next, input how many of that drink you plan to serve. The necessary ingredients are added by the program to the sub-totals for each ingredient.

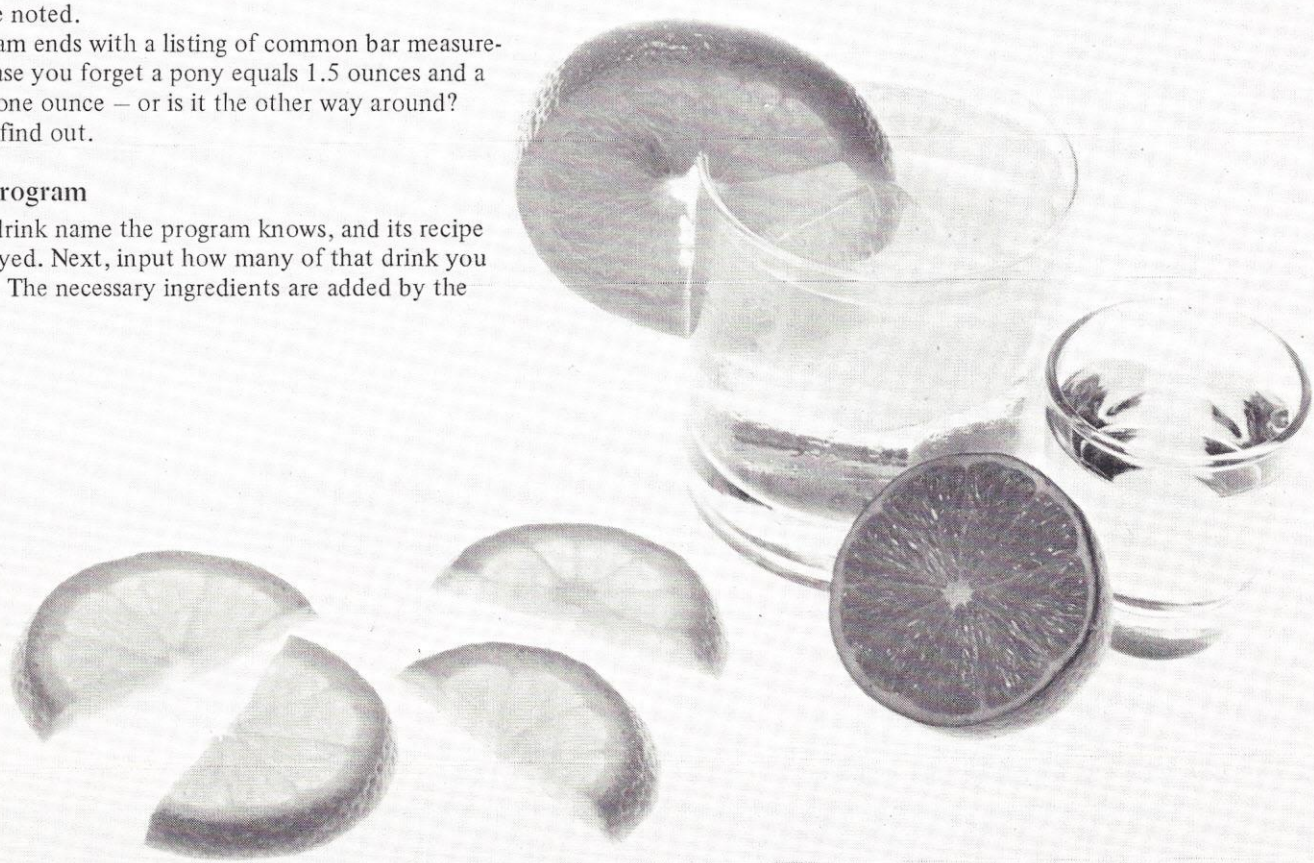
Or, you can name a specific spirit or liqueur you want to use and the program will display all recipes that use that ingredient. Again, you will be asked how many of each drink you want to serve, and the necessary quantity is added to the appropriate ingredient sub-totals.

When you finish choosing recipes, a "shopping list" will show how much of each ingredient you will need to make all the drinks you specified.

Also included in the program is the proper glass, and its capacity, for serving a particular drink.

This program was written using "read" and "data" statements in BASIC. No unusual BASIC commands are used. The interpreter used is MITS 3.4 Disk Extended BASIC. However, no disk or extended commands are used.

Before explaining the structure and use of the "Barkeep program", a review of the "read", "data" and "restore" statements is in order.



Photograph by Steve Krongard

BASIC contains something called a "data pointer". The data pointer points to the next piece of data to be read if the program encounters a "read" statement.

The "restore" statement sets the data pointer to the first data entry of the first "data" statement in the program.

The first "read" statement takes the value of the first data entry and assigns it to the argument of the "read" statement.

For example, the lines listed below would assign the data pointed to by the data pointer to A$.

```
 99 RESTORE
100 READ A$
105 GO TO 100
107 PRINT A$
110 DATA CAR, BOAT, PLANE, TRAIN
```

The first time through the loop, A$ would be assigned the value of "car". The next time, A$ would equal "boat", and so on until the fifth loop, when no data would be left, some error message would be printed out, and the program would stop.

Any program which uses data must check to be sure it has not run out of data in some manner.

Generally, you can solve this problem by structuring your data in a fixed manner, and by using special data termination values. For example:

```
100 READ A$
110 IF A$ = "END" THEN 200
120 READ B
130 PRINT "CAR - "; A$; "YEAR - "; B
140 GO TO 100
150 DATA THUNDERBIRD, 1965, GRAN TORINO, 1970,
    CADILLAC
1969, END
200 STOP
```

In this segment, line 100 first inputs the name of the car; this "name" could be "end" if the data is all used up. This possibility is checked by Line 110. Line 120 reads the year the car was purchased from the data statement.

Line 150 contains the actual data. This line illustrates the use of a data structure and a special data termination value.

Note in the data statement the name of the car comes first followed by the year purchased. Then this pattern is repeated until three cars have been described. Next, following these three data records is the special data termination value, which I arbitrarily call "end".

The data in the Barkeep program consists of drink recipes, names of drinks and names of ingredients. Also included, but not handled in "data" statements, is information on glassware and bar measurements.

Each recipe uses exactly the same data structure. Each recipe starts with its recipe number, which is only used internally by the program. Its purpose is, basically, to save time. When you enter the name of a drink you want to make, that name is compared against the list of stored drink names to see if it's valid. If the program finds the specified drink name in the list, then the program knows its recipe number, and can then retrieve its recipe.

Following the recipe number is the recipe name. After the name is the number of ingredients in the drink.

After the number of ingredients comes the quantity of each ingredient followed by its ingredient number.

Then the recipe itself is listed, enclosed in quotes. (Enclosing the recipe in quotes allows you to use any characters, including commas, in the body of the recipe.)

Finally, each recipe ends with the data entry "end". Also, the last data entry in the whole program is "end". This second "end" is used to detect when all recipes have been read.

Thus, the data structure of recipes: Recipe number, recipe name, number of ingredients in recipe, quantity of first ingredient, ingredient # of first ingredient, quantity of second ingredient, ingredient # of second ingredient, . . . , recipe, "end".

To add a recipe to the program, first check that all of its ingredients are included between Lines 11 - 75. If not, add the new ingredient(s) to the end of the list.

For example, if you wanted to add applejack to the list of ingredients you would first add the following line:

```
76 PRINT "APPLEJACK",: RETURN
```

Next, you would add a "76" to the end of the list of numbers in Line 3030. Finally, you would add "applejack" to the end of the list of ingredients in the data in Lines 6900 - 6919. Also, you must adjust Lines 1006 and 6025 to reflect the total number of ingredients in the data in Lines 6900 - 6919.

To add a new recipe, assuming that all of its ingredients are in the list between Lines 11 - 75, use the preceding recipe data structure to construct data statements to put after the last recipe already included. At present, the next recipe would be put in a data statement after Line 8415.

To calculate the ingredient number, count down from Line 11 starting at "1". For example, curacao is ingredient 1, coffee liqueur is ingredient 5, and creme de cassis is ingredient 10.

Sample Run Notes

In Sample Run 1, I requested recipes for Bloody Marys and Daquiris. Then I requested recipes of drinks using gin. The computer responded with recipes for a Negroni and a Zaza. After I requested a drawing of a cocktail glass, I ended the program and received a complete list of ingredients needed to make the drinks I specified along with a chart of bar measurements.

In Sample Run 2, I specified a drink name, "test", not on the program's list. The computer responded that the drink was unknown to it. Then I asked the program to list all drinks it knew. Next I specified tequilla as my ingredient of choice and was given recipes for Margheritas and Tequilla Sours. The run concluded with a shopping list and bar measurements chart.

Sample Run 3 consists mainly of pictures of each of the nine glasses the program can print. □

SAMPLE RUN 1

```
BERT THE BARTENDER:

SUGGESTIONS:
     FIGURE 2-3 COCKTAILS PER PERSON
     1-2 AFTER DINNER DRINKS PER PERSON
NOTE- ALL MEASUREMENTS ARE IN OUNCES

UNLESS OTHERWISE SPECIFIED.

OPTIONS:
1. MIX A DRINK
2. MIX DRINK WITH A PARTICULAR  INGREDIENT
3. LIST ALL DRINKS THAT I KNOW
4. LIST OF LIQUID MEASURES
5. TIPS ON GLASSWARE
6. END THE PROGRAM
 (PRINT LIST OF ALL REQUIRED INGREDIENTS)

OPTION #? 1
```

```
DRINK NAME? BLOODY MARY

BLOODY MARY
 1.5   VODKA
 3     TOMATO JUICE
 .5    LEMON JUICE
 1     CATSUP (TSP)
 1     WORCESTERSHIRE SAUCE
 1     TABASCO SAUCE (DASH)

SHAKE ALL INGREDIENTS WELL WITH ICE. STRAIN
INTO TALL OR SQUAT 8-OZ. GLASS.
HOW MANY OF THIS DRINK DO YOU PLAN TO
  SERVE? 10
WANT TO SERVE ANOTHER DRINK? YES
DRINK NAME? DAIQUIRI

DAIQUIRI
 2     RUM
 .5    LEMON JUICE
 .5    SUGAR (TSP)

SHAKE WELL WITH ICE. POUR INTO PRECHILLED
SUGAR-FROSTED COCKTAIL GLASS OR OVER THE
ROCKS IN AN OLD-FASHIONED GLASS. SUGAR MAY
BE INCREASED FOR A SWEETER DAIQUIRI.
HOW MANY OF THIS DRINK DO YOU PLAN TO
SERVE? 20

WANT TO SERVE ANOTHER DRINK? NO



OPTIONS:
1. MIX A DRINK
2. MIX DRINK WITH A PARTICULAR  INGREDIENT
3. LIST ALL DRINKS THAT I KNOW
4. LIST OF LIQUID MEASURES
5. TIPS ON GLASSWARE
6. END THE PROGRAM
   (PRINT LIST OF ALL REQUIRED INGREDIENTS)

OPTION #? 2
WHICH SPIRIT OR LIQUEUR DO YOU WANT TO USE?
  GIN

NEGRONI
 .75   CAMPARI
 .75   GIN
 .75   SWEET VERMOUTH

STIR WELL WITH ICE. STRAIN INTO PRECHILLED
COCKTAIL GLASS. MAY BE SERVED ON THE ROCKS
WITH A TWIST OF LEMON OR SPLASH OF SODA OR
BOTH.

HOW MANY OF THIS DRINK DO YOU PLAN TO
  SERVE? 10
WANT TO SEE ANOTHER DRINK WITH SAME
  INGREDIENT? YES

ZAZA
 2     DUBONNET
 1     GIN
 1     ORANGE SLICE

STIR DUBONNET AND GIN WELL WITH ICE.
STRAIN OVER ROCKS IN PRECHILLED
OLD-FASHIONED GLASS. CUT ORANGE
SLICE IN HALF AND PLACE ON THE ROCKS.
YOUR NOSE SHOULD CATCH THE AROMA OF THE
ORANGE BEFORE YOUR LIPS MEET THE DRINK.
HOW MANY OF THIS DRINK DO YOU PLAN TO
  SERVE? 20
WANT TO SEE ANOTHER DRINK WITH SAME
  INGREDIENT? NO
WANT TO SERVE ANY OTHER DRINKS? NO

OPTIONS:
1. MIX A DRINK
2. MIX DRINK WITH A PARTICULAR  INGREDIENT
3. LIST ALL DRINKS THAT I KNOW
4. LIST OF LIQUID MEASURES
5. TIPS ON GLASSWARE
6. END THE PROGRAM
   (PRINT LIST OF ALL REQUIRED INGREDIENTS)

OPTION #? 5
BAR GLASSWARE

1. HIGHBALL OR COLLINS
2. CORDIAL
3. SHOT OR JIGGER
4. DELMONICO OR SOUR
5. WINE GLASS
6. COCKTAIL GLASS
7. CHAMPAGNE GLASS
8. OLD FASHIONED
9. COOLER

ENTER NUMBER OF GLASS YOU WANT  INFO ABOUT.
IF YOU WANT NO MORE INFO, TYPE  'RETURN'? 6

          X          X
          X          X
          X          X   COCKTAIL
          X          X   3-5 OZS.
          X          X
           X        X
            X      X
             XXXXXX
               XX
               XX
               XX
               XX
            ----------
?
BAR GLASSWARE

1. HIGHBALL OR COLLINS
2. CORDIAL
3. SHOT OR JIGGER
4. DELMONICO OR SOUR
5. WINE GLASS
6. COCKTAIL GLASS
7. CHAMPAGNE GLASS
8. OLD FASHIONED
9. COOLER

ENTER NUMBER OF GLASS YOU WANT  INFO ABOUT.
IF YOU WANT NO MORE INFO, TYPE  'RETURN'?
OPTIONS:
1. MIX A DRINK
2. MIX DRINK WITH A PARTICULAR  INGREDIENT
3. LIST ALL DRINKS THAT I KNOW
4. LIST OF LIQUID MEASURES
5. TIPS ON GLASSWARE
6. END THE PROGRAM
   (PRINT LIST OF ALL REQUIRED INGREDIENTS)

OPTION #? 6

TOTAL INGREDIENTS NEEDED
TO MAKE THE DRINKS YOU SPECIFIED:

 7.5   CAMPARI
 7.5   SWEET VERMOUTH
 40    DUBONNET
 40    RUM
 15    VODKA
 27.5  GIN
 15    LEMON JUICE
 10    SUGAR (TSP)
 30    TOMATO JUICE
 10    WORCESTERSHIRE SAUCE
 10    TABASCO SAUCE (DASH)
 10    CATSUP (TSP)
 20    ORANGE SLICE

END OF TABLE OF INGREDIENTS

BAR MEASUREMENTS
DASH- 1/8 OF A TEASPOON
TEASPOON- 1/3 TABLESPOON,1/6 OZ
TABLESPOON- 3 TEASPOONS,.5 OZ.
PONY- 1 OUNCE
JIGGER- 1.5 OUNCES
SPLIT- 6.5 OUNCES
PINT- 16 OUNCES,.5 QUART,2 CUPS
FIFTH- 25.6 OUNCES, 4/5 QUART,  1/5 GALLON
QUART-32 OUNCES,2 PINTS,4 CUPS, 1/4 GALLON
HALF GALLON- 64 OUNCES
MAGNUM- 64 OUNCES
JEROBOAM- 104 OUNCES
?

OK
```

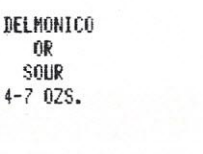
SAMPLE RUN 2

```
BERT THE BARTENDER:

SUGGESTIONS:
     FIGURE 2-3 COCKTAILS PER PERSON
     1-2 AFTER DINNER DRINKS PER PERSON
NOTE- ALL MEASUREMENTS ARE IN OUNCES
UNLESS OTHERWISE SPECIFIED.

OPTIONS:
1. MIX A DRINK
2. MIX DRINK WITH A PARTICULAR  INGREDIENT
3. LIST ALL DRINKS THAT I KNOW
4. LIST OF LIQUID MEASURES
5. TIPS ON GLASSWARE
6. END THE PROGRAM
   (PRINT LIST OF ALL REQUIRED INGREDIENTS)

OPTION #? 1
DRINK NAME? TEST
THAT DRINK IS UNKNOWN TO ME.
OPTIONS:
1. MIX A DRINK
2. MIX DRINK WITH A PARTICULAR  INGREDIENT
3. LIST ALL DRINKS THAT I KNOW
4. LIST OF LIQUID MEASURES
5. TIPS ON GLASSWARE
6. END THE PROGRAM
   (PRINT LIST OF ALL REQUIRED INGREDIENTS)

OPTION #? 3
CHIQUITA PUNCH
GOLD CADILLAC
GRASSHOPPER
LIMEY
ORANGE COMFORT
RUSSIAN COFFEE
AMERICANO
DIABOLO
NEGRONI
VERMOUTH CASSIS
ZAZA
APPLE GRAND MARNIER
BOMBAY
```

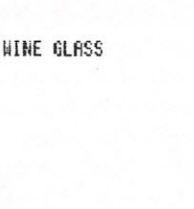
```
FEMINA
STINGER
?
CHAMPAGNE FRAISE
CHAMPAGNE COCKTAIL
BRONX
FOGGY DAY
GIMLET
MARTINI
FROZEN ORANGE BLOSSOM
PINK LADY
BACARDI
CHERRY DAIQUIRI
DAIQUIRI
FROZEN APPLE DAIQUIRI
MAI TAI
SCORPION
MARGERITA
?
TEQUILA SOUR
BLACK RUSSIAN
BLOODY MARY
SCREWDRIVER
SALTY DOG
MANHATTAN
OLD FASHIONED
RUSTY NAIL
TOM COLLINS
BYRRH CASSIS COOLER
MINT JULEP
GIN FIZZ
?
OPTIONS:
1. MIX A DRINK
2. MIX DRINK WITH A PARTICULAR  INGREDIENT
3. LIST ALL DRINKS THAT I KNOW
4. LIST OF LIQUID MEASURES
5. TIPS ON GLASSWARE
6. END THE PROGRAM
   (PRINT LIST OF ALL REQUIRED INGREDIENTS)

OPTION #? 2
WHICH SPIRIT OR LIQUEUR DO YOU WANT TO
  USE? TEQUILLA
MARGERITA
 1.5   TEQUILLA
 .5    TRIPLE SEC
 .5    LEMON JUICE

SHAKE WELL WITH ICE. STRAIN INTO PRE
  CHILLED SALT-RIMMED COCK
TAIL GLASS. TO PREPARE GLASS, RUB RIM WITH
  OUTSIDE OF LEMON
PEEL; THEN DIP INTO SALT AND SHAKE OFF
  EXCESS.
HOW MANY OF THIS DRINK DO YOU PLAN TO
  SERVE? 25
WANT TO SEE ANOTHER DRINK WITH SAME
  INGREDIENT? YES
TEQUILA SOUR
 2     TEQUILLA
 .5    LEMON JUICE
 1     SUGAR (TSP)
 .5    LEMON SLICE

SKAKE TEQUILA, LEMON JUICE, AND SUGAR
  WELL WITH ICE. STRAIN
INTO PRECHILLED WHISKEY-SOUR GLASS. ADD
  LEMON SLICE.
HOW MANY OF THIS DRINK DO YOU PLAN TO
  SERVE? 25
WANT TO SEE ANOTHER DRINK WITH SAME
  INGREDIENT? NO
WANT TO SERVE ANY OTHER DRINKS? NO
OPTIONS:
1. MIX A DRINK
2. MIX DRINK WITH A PARTICULAR  INGREDIENT
3. LIST ALL DRINKS THAT I KNOW
4. LIST OF LIQUID MEASURES
5. TIPS ON GLASSWARE
6. END THE PROGRAM
   (PRINT LIST OF ALL REQUIRED INGREDIENTS)

OPTION #? 6

TOTAL INGREDIENTS NEEDED
TO MAKE THE DRINKS YOU SPECIFIED:

 12.5  TRIPLE SEC
 87.5  TEQUILLA
 25    LEMON JUICE
 25    SUGAR (TSP)
 12.5  LEMON SLICE

END OF TABLE OF INGREDIENTS


BAR MEASUREMENTS
DASH- 1/8 OF A TEASPOON
TEASPOON- 1/3 TABLESPOON;1/6 OZ
TABLESPOON- 3 TEASPOONS;.5 OZ.
PONY- 1 OUNCE
JIGGER- 1.5 OUNCES
SPLIT- 6.5 OUNCES
PINT- 16 OUNCES;.5 QUART;2 CUPS
FIFTH- 25.6 OUNCES; 4/5 QUART;  1/5 GALLON
QUART-32 OUNCES;2 PINTS;4 CUPS; 1/4 GALLON
HALF GALLON- 64 OUNCES
MAGNUM- 64 OUNCES
JEROBOAM- 104 OUNCES
?

OK
```

SAMPLE RUN 3

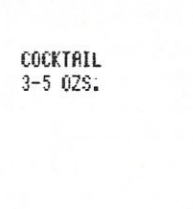
```
BERT THE BARTENDER:

SUGGESTIONS:
     FIGURE 2-3 COCKTAILS PER PERSON
     1-2 AFTER DINNER DRINKS PER PERSON
NOTE- ALL MEASUREMENTS ARE IN OUNCES
UNLESS OTHERWISE SPECIFIED.


OPTIONS:
1. MIX A DRINK
2. MIX DRINK WITH A PARTICULAR  INGREDIENT
3. LIST ALL DRINKS THAT I KNOW
4. LIST OF LIQUID MEASURES
5. TIPS ON GLASSWARE
6. END THE PROGRAM
   (PRINT LIST OF ALL REQUIRED INGREDIENTS)


OPTION #? 5
BAR GLASSWARE

1. HIGHBALL OR COLLINS
2. CORDIAL
3. SHOT OR JIGGER
4. DELMONICO OR SOUR
5. WINE GLASS
6. COCKTAIL GLASS
7. CHAMPAGNE GLASS
8. OLD FASHIONED
9. COOLER

ENTER NUMBER OF GLASS YOU WANT  INFO ABOUT.
IF YOU WANT NO MORE INFO; TYPE  'RETURN'? 1

      X     X
      X     X
      X     X
      X     X   HIGHBALL
      X     X     OR
      X     X   COLLINS
      X     X    8-11
      X     X    OZS.
      X     X
      X     X
      X     X
       -----

?
BAR GLASSWARE

1. HIGHBALL OR COLLINS
2. CORDIAL
3. SHOT OR JIGGER
4. DELMONICO OR SOUR
5. WINE GLASS
6. COCKTAIL GLASS
7. CHAMPAGNE GLASS
8. OLD FASHIONED
9. COOLER

ENTER NUMBER OF GLASS YOU WANT  INFO ABOUT.
IF YOU WANT NO MORE INFO; TYPE  'RETURN'? 2

      X   X
      X   X
      X   X   CORDIAL
      X   X    1 OZ.
       X X
        X
        X
        X
       -----

?
BAR GLASSWARE

1. HIGHBALL OR COLLINS
2. CORDIAL
3. SHOT OR JIGGER
4. DELMONICO OR SOUR
5. WINE GLASS
6. COCKTAIL GLASS
7. CHAMPAGNE GLASS
8. OLD FASHIONED
9. COOLER

ENTER NUMBER OF GLASS YOU WANT  INFO ABOUT.
IF YOU WANT NO MORE INFO; TYPE  'RETURN'? 3

      X     X
      X     X
       X   X    SHOT OR JIGGER
        XXX
        XXX       1.5 OZS.
        XXX
       -----

1. HIGHBALL OR COLLINS
2. CORDIAL
3. SHOT OR JIGGER
4. DELMONICO OR SOUR
5. WINE GLASS
6. COCKTAIL GLASS
7. CHAMPAGNE GLASS
8. OLD FASHIONED
9. COOLER

ENTER NUMBER OF GLASS YOU WANT  INFO ABOUT.
IF YOU WANT NO MORE INFO; TYPE  'RETURN'? 4
```




```
?
BAR GLASSWARE

1. HIGHBALL OR COLLINS
2. CORDIAL
3. SHOT OR JIGGER
4. DELMONICO OR SOUR
5. WINE GLASS
6. COCKTAIL GLASS
7. CHAMPAGNE GLASS
8. OLD FASHIONED
9. COOLER

ENTER NUMBER OF GLASS YOU WANT  INFO ABOUT.
IF YOU WANT NO MORE INFO, TYPE  'RETURN'? 5
```



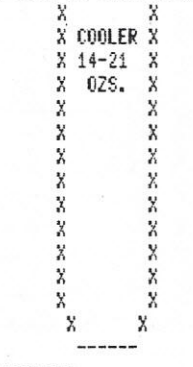

```
?
BAR GLASSWARE

1. HIGHBALL OR COLLINS
2. CORDIAL
3. SHOT OR JIGGER
4. DELMONICO OR SOUR
5. WINE GLASS
6. COCKTAIL GLASS
7. CHAMPAGNE GLASS
8. OLD FASHIONED
9. COOLER

ENTER NUMBER OF GLASS YOU WANT  INFO ABOUT.
IF YOU WANT NO MORE INFO, TYPE  'RETURN'? 6
```




```
?
BAR GLASSWARE
      \
1. HIGHBALL OR COLLINS
2. CORDIAL
3. SHOT OR JIGGER
4. DELMONICO OR SOUR
5. WINE GLASS
6. COCKTAIL GLASS
7. CHAMPAGNE GLASS
8. OLD FASHIONED
9. COOLER

ENTER NUMBER OF GLASS YOU WANT  INFO ABOUT.
IF YOU WANT NO MORE INFO, TYPE  'RETURN'? 7
```


```
?
BAR GLASSWARE

1. HIGHBALL OR COLLINS
2. CORDIAL
3. SHOT OR JIGGER
4. DELMONICO OR SOUR
5. WINE GLASS
6. COCKTAIL GLASS
7. CHAMPAGNE GLASS
8. OLD FASHIONED
9. COOLER

ENTER NUMBER OF GLASS YOU WANT  INFO ABOUT.
IF YOU WANT NO MORE INFO, TYPE  'RETURN'? 8
```


```
?
BAR GLASSWARE

1. HIGHBALL OR COLLINS
2. CORDIAL
3. SHOT OR JIGGER
4. DELMONICO OR SOUR
5. WINE GLASS
6. COCKTAIL GLASS
7. CHAMPAGNE GLASS
8. OLD FASHIONED
9. COOLER

ENTER NUMBER OF GLASS YOU WANT  INFO ABOUT.
IF YOU WANT NO MORE INFO, TYPE  'RETURN'? 9
```




```
?
BAR GLASSWARE

1. HIGHBALL OR COLLINS
2. CORDIAL
3. SHOT OR JIGGER
4. DELMONICO OR SOUR
5. WINE GLASS
6. COCKTAIL GLASS
7. CHAMPAGNE GLASS
8. OLD FASHIONED
9. COOLER

ENTER NUMBER OF GLASS YOU WANT  INFO ABOUT.
IF YOU WANT NO MORE INFO, TYPE  'RETURN'?
OPTIONS:
1. MIX A DRINK
2. MIX DRINK WITH A PARTICULAR  INGREDIENT
3. LIST ALL DRINKS THAT I KNOW
4. LIST OF LIQUID MEASURES
5. TIPS ON GLASSWARE
6. END THE PROGRAM
   (PRINT LIST OF ALL REQUIRED INGREDIENTS)

OPTION #? 1
DRINK NAME? OLD

OLD FASHIONED
 .5    SUGAR (TSP)
 2     ANGOSTURA BITTERS (DASH)
 2     CLUB SODA
 2     BLENDED WHISKEY
 1     LEMON PEEL

STIR
SUGAR, BITTERS AND SODA WATER IN PRECHILLED
OLD-FASHION ED GLASS UNTIL SUGAR DISSOLVES.
FILL GLASS WITH ICE CUBES OR LARGE PIECES OF
CRACKED ICE. ADD WHISKEY. STIR WELL. TWIST
LEMON PEEL ABOVE DRINK AND DROP INTO GLASS.
HOW MANY OF THIS DRINK DO YOU PLAN TO
SERVE? 10
WANT TO SERVE ANOTHER DRINK? NO
OPTIONS:
1. MIX A DRINK
2. MIX DRINK WITH A PARTICULAR  INGREDIENT
3. LIST ALL DRINKS THAT I KNOW
4. LIST OF LIQUID MEASURES
5. TIPS ON GLASSWARE
6. END THE PROGRAM
   (PRINT LIST OF ALL REQUIRED INGREDIENTS)

OPTION #? 6

TOTAL INGREDIENTS NEEDED
TO MAKE THE DRINKS YOU SPECIFIED:

 20    BLENDED WHISKEY
 20    CLUB SODA
 20    ANGOSTURA BITTERS (DASH)
 5     SUGAR (TSP)
 10    LEMON PEEL

END OF TABLE OF INGREDIENTS



BAR MEASUREMENTS
DASH- 1/8 OF A TEASPOON
TEASPOON- 1/3 TABLESPOON,1/6 OZ
TABLESPOON- 3 TEASPOONS,.5 OZ.
PONY- 1 OUNCE
JIGGER- 1.5 OUNCES
SPLIT- 6.5 OUNCES
PINT- 16 OUNCES,.5 QUART,2 CUPS
FIFTH- 25.6 OUNCES, 4/5 QUART,  1/5 GALLON
QUART-32 OUNCES,2 PINTS,4 CUPS, 1/4 GALLON
HALF GALLON- 64 OUNCES
MAGNUM- 64 OUNCES
JEROBOAM- 104 OUNCES
?

OK
```

PROGRAM NOTES

The bulk of the program consists of data, either stored in "data" statements or as "print" statements. Lines 1 - 9 are for initialization.

In Line 5, all of the subscripted variables are declared. IM(I) is the variable which stores all of the ingredient subtotals. For example, IM(5) would contain the number of units of ingredient 5 (coffee liqueur) needed to make all of the drinks specified. IN(I) and Q(I) are used to temporarily store each recipe as it is displayed. IN(I) contains the ingredient used and Q(I) contains the quantity of each of those ingredients needed to make one drink.

For example, in Drink 1, chiquita punch, Line 8000, IN(I) is 29, Q(I) is 1.5, IN(2) is 42, Q(2) is 1.5, and so forth.

The first number after the name of each drink, in this case "5", is the number of different ingredients needed to make the drink.

Lines 11 - 75 contain the names of all the ingredients in the form of "print" statements, in order of ingredient number.

Ignore Line 1000 – it is never used.

From Line 1005 - 1110 is a subroutine which, given a recipe number, searches for and displays that recipe, storing its recipe in IN(I) and Q(I). Then it asks you how many drinks of this kind you want.

The preceding subroutine is entered from Line 3790, which in turn is the end of a routine which asks you what drink you want and then sees if that drink is on the list of drinks. If the drink you specify is on the list, its recipe number is passed to variable "D", which is used by the preceding routine (Lines 1005 - 1110) to retrieve the proper recipe.

Lines 1500 - 1560 are a routine which asks for an ingredient name and calculates its number, if that ingredient is on the stored list of ingredients in Lines 6900 - 6919.

Lines 2000 - 2010 are a routine which simply skips through a recipe if it does not have the recipe number we are looking for. This routine is called by Line 1015. The routine returns, when finished skipping a recipe, to Line 1010, which then reads the next recipe number and compares it in Line 1015 to see if it is the one we are looking for.

Line 3000 - 3040 are a routine which gets an ingredient number and prints that ingredient name, using Lines 11 - 75. This routine (3000 - 3040) is called from many points in the program.

Lines 3500 - 3570 list the drink names that are in the list in Lines 7000 - 7080. You must adjust Line 3525 to reflect the number of ingredients in Lines 6900 - 6919.

This routine is formatted for a SWTPC-1024 Terminal. The question marks, produced by the input statements in Lines 3540 and 3560, are designed to stop the input so that a full screen of information is displayed. When you type "return", the screen is erased and new information is displayed to fill the screen.

Lines 3700 - 3790 are the routine which inputs drink name and returns its recipe number described above.

Lines 4000 - 4140 print out the bar measurements.

Lines 4150 - 4180 are executed right after initialization. They are opening remarks and comments. They also serve to identify the program.

Lines 4200 - 4320 are the instruction decoder. This routine displays an options menu and requests you enter your choice. The options are : 1. Mix a drink, 2. mix a drink or drinks with a specified liqueur, 3. list all drinks, 4. print out bar measurements, 5. print outlines and capacities of popular bar glassware, and 6. print "shopping list" of ingredients needed to make all the drinks you specified.

Lines 5000 - 5150 print a menu of popular bar glassware and give you a choice of which glass to display.

Lines 6000 -6270 will ask you what ingredient you want to use, and will display any or all recipes that use that ingredient. Lines 6025 and 6027 must be changed if the number of ingredients or drinks is changed. These two lines skip over the data in Lines 6900 - 7080.

Lines 6900 - 6919 are data statements containing names of spirits and liqueurs.

Lines 7000 - 7080 are data statements containing names of all drinks.

Lines 8000 - 8415 are data statements containing all of the recipes. Each recipe uses the same recipe data structure, described above.

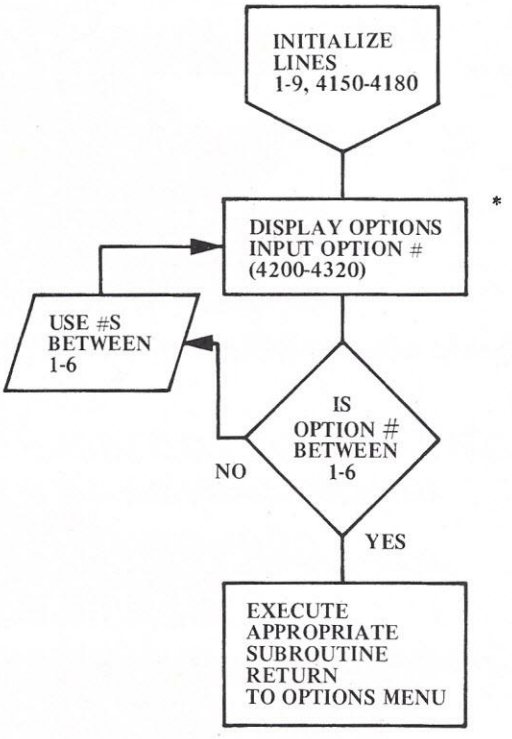
Lines 9000 - 9990 print outline drawings and capacities of popular bar glassware. These drawings are formatted for a SWTPC-1024 screen. "Input A$" in Lines 9110, 9195, 9270, 9370, 9470, 9590, 9750, 9890 and 9990 stops the computer so that a stable image remains on the screen. When you type "return", the image of the glass is erased, and a new menu is printed on the screen.

Lines 10000 - 10100 print you a shopping list of all ingredients necessary to make the drinks you specified while running the program. Adjust Line 10050 if the number of ingredients changes. All units are in ounces unless otherwise specified.

PROGRAM LISTING

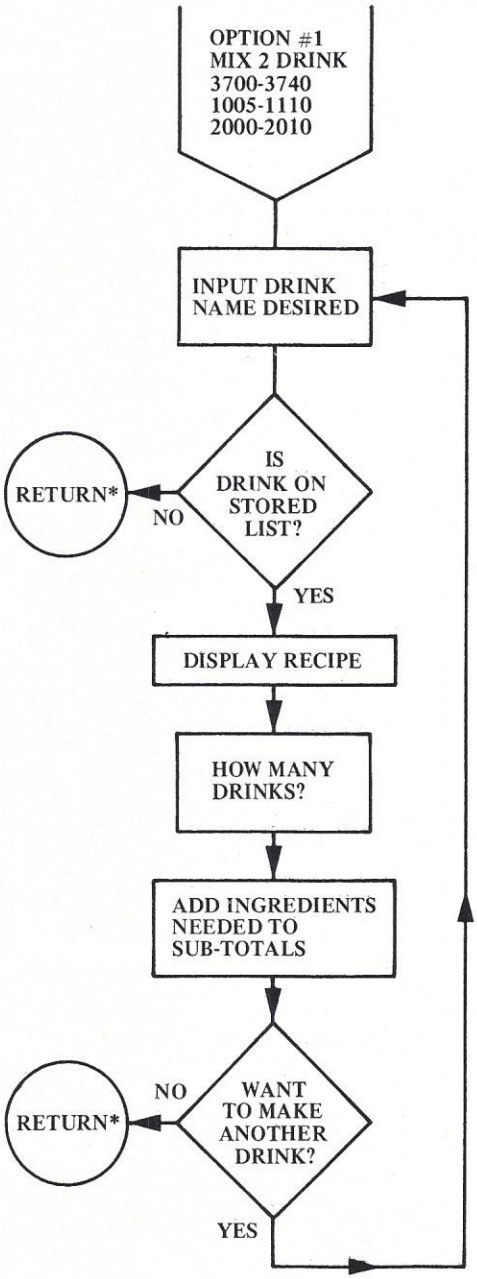
```
1 REM *** BARKEEP PROGRAM ***
2 REM *** BY SAM NEWHOUSE
3 REM *** COPYRIGHT 1978 BY SAM NEWHOUSE
4 PRINT:PRINT
5 CLEAR 2000:DIM IM(100),IN(100),Q(10)
6 PRINT"BERT THE BARTENDER:":PRINT
8 WIDTH 230
9 GOTO 4150
11 PRINT"CURACAO";:RETURN
12 PRINT"BRANDY";:RETURN
13 PRINT"TRIPLE SEC";:RETURN
14 PRINT"GALLIANO";:RETURN
15 PRINT"COFFEE LIQUER";:RETURN
16 PRINT"CHERRY LIQUER";:RETURN
17 PRINT"CREME DE CACAO";:RETURN
18 PRINT"WHITE CREME DE MENTHE";:RETURN
19 PRINT"ANISETTE";:RETURN
20 PRINT"CREME DE CASSIS";:RETURN
21 PRINT"LIME LIQUER";:RETURN
22 PRINT"KIRSCH";:RETURN
23 PRINT"COMPARI";:RETURN
24 PRINT"SWEET VERMOUTH";:RETURN
25 PRINT"DRY VERMOUTH";:RETURN
26 PRINT"DUBONNET";:RETURN
27 PRINT"PORT";:RETURN
28 PRINT"COGNAC";:RETURN
29 PRINT"GRAND MARNIER";:RETURN
30 PRINT"CALVADOS";:RETURN
31 PRINT"DRAMBUIE";:RETURN
32 PRINT"BENEDICTINE";:RETURN
33 PRINT"PERNOD";:RETURN
34 PRINT"STRAWBERRY LIQUER";:RETURN
35 PRINT"BRUT CHAMPAGNE";:RETURN
36 PRINT"ORZATA";:RETURN
37 PRINT"KAHLUA";:RETURN
38 PRINT"BYRRH";:RETURN
39 PRINT"BANNANA LIQUER";:RETURN
40 PRINT"RUM";:RETURN
41 PRINT"VODKA";:RETURN
42 PRINT"GIN";:RETURN
43 PRINT"SOUTHERN COMFORT";:RETURN
44 PRINT"BLENDED WHISKEY";:RETURN
45 PRINT"SCOTCH";:RETURN
46 PRINT"BOURBON";:RETURN
47 PRINT"TEQUILLA";:RETURN
48 PRINT"LEMON JUICE";:RETURN
49 PRINT"CREAM";:RETURN
50 PRINT"EGG WHITE";:RETURN
51 PRINT"CRACKED ICE (CUP)";:RETURN
52 PRINT"ORANGE JUICE";:RETURN
53 PRINT"GRENADINE (TSP)";:RETURN
54 PRINT"LIME JUICE";:RETURN
55 PRINT"CLUB SODA";:RETURN
56 PRINT"ANGOSTURA BITTERS (DASH)";:RETURN
57 PRINT"SUGAR (TSP)";:RETURN
58 PRINT"GRAPEFRUIT JUICE";:RETURN
59 PRINT"ORANGE-FLOWER WATER ";:RETURN
60 PRINT"APPLE JUICE";:RETURN
61 PRINT"TOMATO JUICE";:RETURN
62 PRINT"WORCESTERSHIRE SAUCE ";:RETURN
63 PRINT"TABASCO SAUCE (DASH)";:RETURN
64 PRINT"CATSUP (TSP)";:RETURN
65 PRINT"ORANGE SLICE";:RETURN
66 PRINT"LIME SLICE";:RETURN
67 PRINT"LEMON SLICE";:RETURN
68 PRINT"LEMON PEEL";:RETURN
69 PRINT"ORANGE PEEL";:RETURN
70 PRINT"STRAWBERRY";:RETURN
71 PRINT"PINEAPPLE CHUNK";:RETURN
72 PRINT"MINT SPRIG";:RETURN
73 PRINT"MINT LEAF";:RETURN
74 PRINT"GREEN CREME DE MENTHE";:RETURN
75 PRINT"APPLE SLICE";:RETURN
1000 PRINTCHR$(16);CHR$(22);:PRINT:INPUT"DRINK #";D
1005 RESTORE
1006 FOR I=1 TO 38:READ A$:NEXTI
1007 FORI=1 TO 43:READ A$:NEXTI
1010 READ DI
1015 IF DI<>D THEN 2000
1020 READ D$:PRINTCHR$(16);CHR$(22);
1025 PRINT:PRINTD$
1030 READ NI
1035 FORI=1 TO NI
1040 READ Q(I),IN(I)
1045 NEXT I
1050 READ RR$
1055 FORI=1 TO NI
1060 PRINT Q(I);TAB(7);:IN=IN(I):GOSUB3000:PRINT
1065 NEXT I
1070 PRINT:PRINTRR$
1075 INPUT"HOW MANY OF THIS DRINK DO YOU PLAN TO SERVE";HM
1080 FOR I=1 TO NI
1085 IM(IN(I))=IM(IN(I))+(HM*Q(I))
1090 NEXT I
1095 YN$="*":INPUT"WANT TO SERVE ANOTHER DRINK";YN$
1100 IF LEFT$(YN$,1)="Y" THEN 3700
1105 IF LEFT$(YN$,1)="N" THEN RETURN
1110 PRINT"USE 'YES' OR 'NO'.":GOTO1095
1500 REM *** CALCULATE INGREDIENT #
1510 RESTORE:IC=1
1520 INPUT"WHICH SPIRIT OR LIQUEUR DO YOU WANT TO USE";SI$
1530 SI=0
1540 READ A$:IF A$="END" THEN PRINT"I DON'T KNOW THAT INGRED
IENT.":GOTO1500
1550 IF LEFT$(A$,LEN(SI$))=SI$ THEN SI=IC:GOTO6000
1560 IC=IC+1:GOTO 1540
2000 READ A$
2005 IF A$="ENB" THEN 1010
```

FIGURE 1



SEE FIGURES 2 to 7
FOR EACH OPTION'S FLOWCHART

FIGURE 2

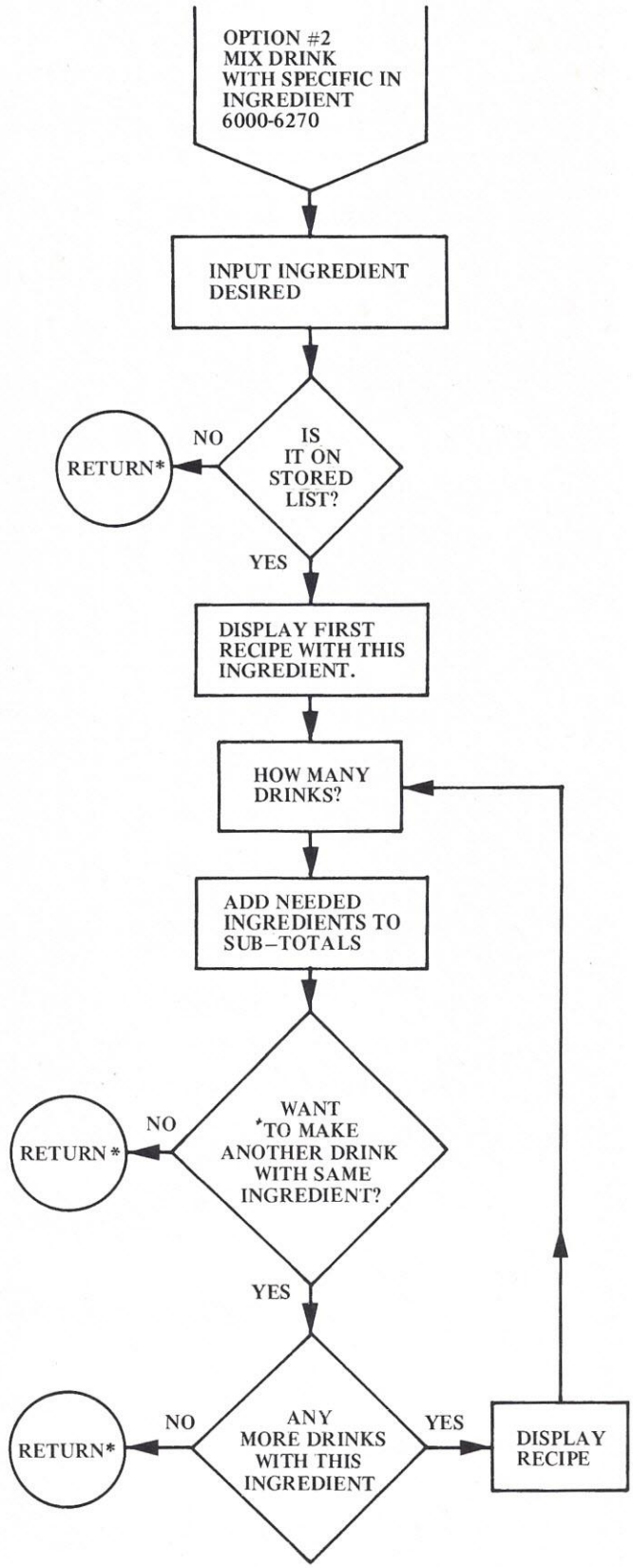


*GO TO POINT MARKED * ON FIGURE 1

```
2010 GOTO 2000
3000 REM *** PRINT INGREDIENT
3010 L=0
3030 ON IN GOSUB11,12,13,14,15,16,17,18,19,20,21,22,23,24,25
,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45
,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65
,66,67,68,69,70,71,72,73,74,75
3040 RETURN
3500 PRINTCHR$(16);CHR$(22);
3510 L=0
3520 RESTORE
3525 FORI=1TO38:READA$:NEXTI
3530 READ A$
3540 IF A$="END" THEN INPUT A$:RETURN
3550 PRINTA$:L=L+1
3560 IF L>14 THEN INPUT A$:L=0:PRINTCHR$(16);CHR$(22);
3570 GOTO 3530
3700 PRINTCHR$(16);CHR$(22);:INPUT"DRINK NAME";DD$
3710 RESTORE:D=1
3715 FORI=1TO38:READ A$:NEXTI
3720 READ A$
3730 IF LEFT$(A$,LEN(DD$))=DD$ THEN 3790
3740 IF A$="END" THEN PRINT"THAT DRINK IS UNKNOWN TO ME.":RE
TURN
3750 D=D+1:GOTO 3720
3790 GOTO1005
4000 REM *** LIST OF MEASUREMENTS
4010 PRINTCHR$(16);CHR$(22);"BAR MEASUREMENTS"
4020 PRINT"DASH- 1/8 OF A TEASPOON
4030 PRINT"TEASPOON- 1/3 TABLESPOON,1/6 OZ
4040 PRINT"TABLESPOON- 3 TEASPOONS,.5 OZ."
4050 PRINT"PONY- 1 OUNCE
4060 PRINT"JIGGER- 1.5 OUNCES"
4070 PRINT"SPLIT- 6.5 OUNCES"
4080 PRINT"PINT- 16 OUNCES,.5 QUART,2 CUPS"
4090 PRINT"FIFTH- 25.6 OUNCES, 4/5 QUART,  1/5 GALLON"
4100 PRINT"QUART-32 OUNCES,2 PINTS,4 CUPS, 1/4 GALLON"
4110 PRINT"HALF GALLON- 64 OUNCES"
4120 PRINT"MAGNUM- 64 OUNCES"
4130 PRINT"JEROBOAM- 104 OUNCES"
4140 INPUT A$:RETURN
4150 PRINT"SUGGESTIONS:"
4155 PRINT"     FIGURE 2-3 COCKTAILS PER PERSON"
4160 PRINT"     1-2 AFTER DINNER DRINKS PER PERSON"
4170 PRINT"NOTE- ALL MEASUREMENTS ARE IN OUNCES"
4175 PRINT"UNLESS OTHERWISE SPECIFIED."
4180 PRINT:PRINT
4200 REM *** MAIN LOOP-INSTRUCTION DECODER
4210 PRINTCHR$(16);CHR$(22);
4220 PRINT"OPTIONS:"
4230 PRINT"1. MIX A DRINK"
4240 PRINT"2. MIX DRINK WITH A PARTICULAR  INGREDIENT"
4250 PRINT"3. LIST ALL DRINKS THAT I KNOW"
4260 PRINT"4. LIST OF LIQUID MEASURES"
4270 PRINT"5. TIPS ON GLASSWARE"
4280 PRINT"6. END THE PROGRAM"
4285 PRINT"   (PRINT LIST OF ALL REQUIRED INGREDIENTS)"
4290 PRINT:INPUT"OPTION #";OP
4300 IF OP<1 OR OP>6 THEN PRINT"USE NUMBERS 1-6.":GOTO 4200
4310 ON OP GOSUB 3700,1500,3500,4000,5000,10000
4320 GOTO 4200
5000 REM *** DISPLAY GLASSWARE
5010 PRINTCHR$(16);CHR$(22);"BAR GLASSWARE"
5020 PRINT:PRINT"1. HIGHBALL OR COLLINS"
5030 PRINT"2. CORDIAL"
5040 PRINT"3. SHOT OR JIGGER"
5050 PRINT"4. DELMONICO OR SOUR"
5060 PRINT"5. WINE GLASS"
5070 PRINT"6. COCKTAIL GLASS"
5080 PRINT"7. CHAMPAGNE GLASS"
5090 PRINT"8. OLD FASHIONED"
5100 PRINT"9. COOLER"
5110 PRINT:PRINT"ENTER NUMBER OF GLASS YOU WANT  INFO ABOUT.
"
5120 G=0:INPUT"IF YOU WANT NO MORE INFO, TYPE  'RETURN'";G
5130 IF G=0 THEN PRINTCHR$(16);CHR$(22);:RETURN
5135 IF G<1 OR G>9 THEN PRINT"USE NUMBERS BETWEEN 1 AND 9.":
GOTO5110
5140 ON G GOSUB 9000,9120,9200,9300,9400,9500,9600,9800,9900
5150 GOTO 5010
6000 REM *** SELECT DRINKS WITH GIVEN INGREDIENT
6010 PRINTCHR$(16);CHR$(22);
6020 RESTORE
6025 FORI=1 TO 38:READ A$:NEXT I
6027 FORI=1 TO 43:READ A$:NEXTI
6030 READ DI$:IF DI$="END" THENPRINT"END OF LIST OF DRINKS":
 RETURN
6035 READ R$,NI
6040 FORI=1TONI
6050 READ Q(I),IN(I)
6060 NEXTI
6070 READ RR$
6080 ST=0:REM ST IS SEARCH TEST FLAG-1 FOR YES, 0 FOR NO
```

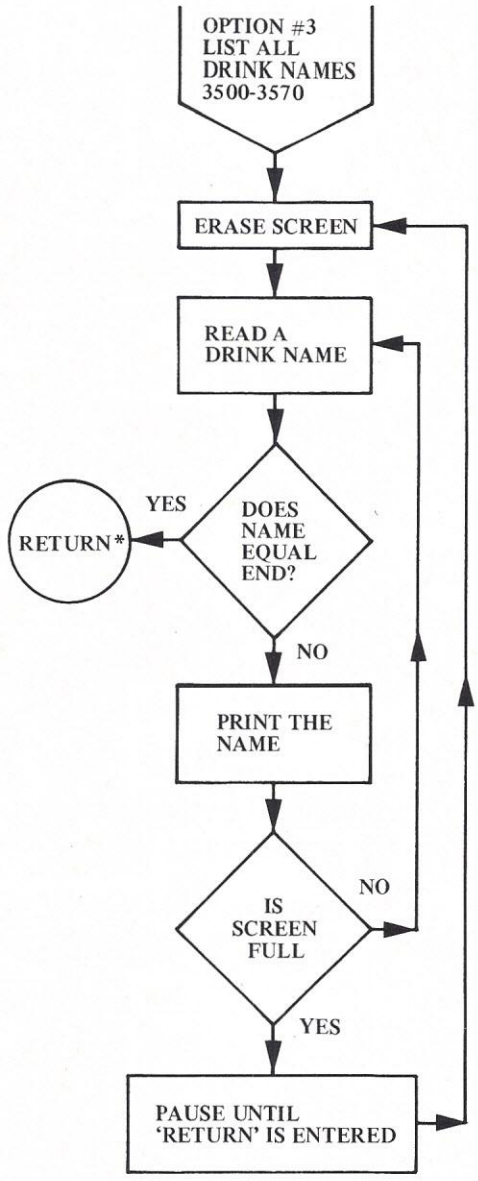
```
6090 FORI=1 TO NI
6100 IF IN(I)=SI THEN ST=1
6110 NEXT I
6120 IF ST=0 THEN READ A$:GOTO6030
6130 READ A$:PRINTR$
6140 FORI=1 TO NI
6150 PRINTQ(I);TAB(7);:IN=IN(I):GOSUB3000:PRINT
6160 NEXTI
6170 PRINT:PRINTRR$
6175 HM=0:INPUT"HOW MANY OF THIS DRINK DO YOU PLAN TO SERVE"
;HM
6180 FORI=1 TO NI
6190 IM(IN(I))=IM(IN(I))+(HM*Q(I))
6195 NEXTI
6200 YN$="YES":INPUT"WANT TO SEE ANOTHER DRINK WITH SAME ING
REDIENT";YN$
6210 IF YN$="YES" THEN 6030
6220 IF YN$="NO" THEN 6240
6230 PRINT"USE 'YES' OR 'NO'.":GOTO 6200
6240 YN$="YES":INPUT"WANT TO SERVE ANY OTHER DRINKS";YN$
6250 IF YN$="YES" THEN 4200
6260 IF YN$="NO" THEN RETURN
6270 PRINT"USE 'YES' OR 'NO'.":GOTO 6240
6900 DATA CURACAO,BRANDY,TRIPLE SEC,GALLIANO,COFFEE LIQUEUR,
CHERRY LIQUEUR,CREME DE CACAO,WHITE CREME DE MENTHE,ANISETTE
,CREME DE CASSIS,LIME LIQUEUR,KIRSCH,COMPARI,SWEET VERMOUTH,
DRY VERMOUTH,DUBONNET,PORT,COGNAC,GRAND MARNIER,CALVADOS
6919 DATA DRAMBUIE,BENEDICTINE,PERNOD,STRAWBERRY LIQUEUR,BRU
T CHAMPAGNE,ORZATA,KAHLUA,BYRRH,BANANA LIQUEUR,RUM,VODKA,GI
N,SOUTHERN COMFORT,BLENDED WHISKEY,SCOTCH,BOURBON,TEQUILLA,E
ND
7000 DATA CHIQUITA PUNCH,GOLD CADILLAC,GRASSHOPPER,LIMEY
7010 DATA ORANGE COMFORT,RUSSIAN COFFEE,AMERICANO,DIABOLO,NE
GRONI
7020 DATA VERMOUTH CASSIS,ZAZA,APPLE GRAND MARNIER,BOMBAY,FE
MINA
7030 DATA STINGER,CHAMPAGNE FRAISE,CHAMPAGNE COCKTAIL,BRONX,
FOGGY DAY
7040 DATA GIMLET,MARTINI,FROZEN ORANGE BLOSSOM,PINK LADY,BAC
ARDI
7050 DATA CHERRY DAIQUIRI,DAIQUIRI,FROZEN APPLE DAIQUIRI,MAI
 TAI,SCORPION
7060 DATA MARGERITA,TEQUILA SOUR,BLACK RUSSIAN,BLOODY MARY,S
CREWDRIVER,SALTY DOG
7070 DATA MANHATTAN,OLD FASHIONED,RUSTY NAIL,TOM COLLINS,BYR
RH CASSIS COOLER,MINT JULEP,GIN FIZZ
7080 DATA "END"
8000 DATA 1,CHIQUITA PUNCH,5,1.5,29,1.5,42,1.5,39,.75,43,.75
,41,"PUT ALL INGREDIENTS INTO BLENDER. BLEND AT HIGH SPEED 1
0 SECONDS. POUR INTO PRE-CHILLED OLD-FASHIONED GLASS.",END
8010 DATA 2,GOLD CADILLAC,4,.75,7,.75,4,.75,39,.33,41,"PUT A
LL INGREDIENTS INTO BLENDER. BLEND AT LOW SPEED 10-15 SECOND
S. POUR INTO PRECHILLED CHAMPAGNE GLASS.",END
8020 DATA 3,GRASSHOPPER,3,.75,7,.75,64,.75,39,"SHAKE WITH IC
E. STRAIN INTO PRECHILLED COCKTAIL GLASS.",END
8030 DATA 4,LIMEY,6,1,30,1,11,.5,3,.33,44,.33,41,1,56
8035 DATA "PUT RUM, LIME LIQUEUR, TRIPLE SEC, LIME JUICE, AN
D CRUSHED ICE INTO BLENDER. BLEND AT LOW SPEED 10-15 SECONDS
. POUR INTO PRECHILLED CHAMPAGNE GLASS. ADD LIME SLICE.",END
8040 DATA 5,ORANGE COMFORT,4,.5,33,.5,9,.75,42,.5,38,"SHAKE
SOUTHERN COMFORT, ANISETTE, ORANGE JUICE, AND LEMON JUICE WE
LL WITH ICE. STRAIN INTO PRECHILLED COCKTAIL GLASS.",END
8050 DATA 6,RUSSIAN COFFEE,4,.75,5,.75,31,.75,39,.33,41,"PUT
 ALL INGREDIENTS INTO BLENDER. BLEND AT LOW SPEED 10-15 SECO
NDS. POUR INTO PRECHILLED CHAMPAGNE GLASS.",END
8060 DATA 7,AMERICANO,4,1.25,13,1.25,14,1,58,4,45
8065 DATA "STIR CAMPARI AND SWEET VERMOUTH WELL WITH ICE. ST
RAIN INTO PRECHILLED COCKTAIL GLASS. TWIST LEMON PEEL ABOVE
DRINK AND DROP INTO GLASS. IF YOU PREFER, A DELMONICO OR OLD
-FASHIONED GLASS MAY BE USED INSTEAD.",END
8070 DATA 8,DIABOLO,4,1.5,17,1,15,.25,38,1,58,"SHAKE PORT, V
ERMOUTH, AND LEMON JUYCE WELL WITH ICE. STRAIN INTO PRECHILL
ED COCKTAIL GLASS. TWIST LEMON PEEL ABOVE DRINK AND DROP INT
O GLASS.",END
8080 DATA 9,NEGRONI,3,.75,13,.75,32,.75,14
8085 DATA "STIR WELL WITH ICE. STRAIN INTO PRECHILLED COCKTA
```

FIGURE 3



*GO TO POINT MARKED * ON FIGURE 1

FIGURE 4

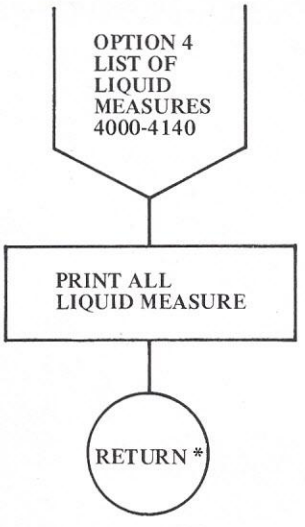


*GO TO SHEET MARKED * ON FIGURE 1

```
IL GLASS. MAY BE SERVED ON THE ROCKS WITH A TWIST OF LEMON O
R SPLASH OF SODA OR BOTH.";END
8090 DATA 10;VERMOUTH CASSIS;3;2;15;1;10;4;45
8095 DATA "POUR VERMOUTH AND CREME DE CASSIS OVER ONE OR TWO
 ROCKS IN A PRECHILLED OLD-FASHIONED GLASS. STIR. ADD SODA.
A SLICE OF LEMON MAY BE USED AS A GARNISH IF DESIRED.";END
8100 DATA 11;ZAZA;3;2;16;1;32;1;55
8105 DATA "STIR DUBONNET AND GIN WELL WITH ICE. STRAIN OVER
ROCKS IN PRECHILLED OLD-FASHIONED GLASS. CUT ORANGE SLICE IN
 HALF AND PLACE ON THE ROCKS. YOUR NOSE SHOULD CATCH THE ARO
MA OF THE ORANGE BEFORE YOUR LIPS MEET THE DRINK.";END
8110 DATA 12;APPLE GRAND MARNIER;5;1;20;.5;19;.5;18;1;58;1;5
9
8115 DATA "STIR CALVADOS, GRAND MARNIER AND COGNAC WELL WITH
 ICE. STRAIN OVER ROCKS IN PRECHILLED OLD-FASHIONED GLASS. T
WIST FRUIT PEELS ABOVE DRINK AND DROP INTO GLASS.";END
8120 DATA 13;BOMBAY;5;1;2;.5;15;.5;14;.1;1;.1;23
8125 DATA "SHAKE BRANDY, BOTH KINDS OF VERMOUTH, CURACAO AND
 PERNOD WELL WITH ICE. STRAIN OVER ROCKS IN PRECHILLED OLD-F
ASHIONED GLASS. SERVE BEFORE A CURRY DINNER.";END
8130 DATA 14;FEMINA;3;1.5;2;.5;22;.5;42;"SHAKE BRANDY, BENED
ICTINE, AND ORANGE JUICE WELL WITH ICE. STRAIN OVER ROCKS IN
 PRECHILLED OLD-FASHIONED GLASS.";END
8140 DATA 15;STINGER;2;1.25;2;1.25;8
8145 DATA"SHAKE WELL WITH ICE. STRAIN INTO PRECHILLED COCKTA
IL GLASS. FOR A DRY STINGER, INCREASE BRANDY TO 2 OZS. AND R
EDUCE CREME DE MENTHE TO .5 OZ. MAY BE OFFERED BEFORE OR AFT
ER DINNER. IT IS FREQUENTLY SERVED WITH A GLASS OF ICE WATER
 ON THE SIDE."
8147 DATA END
8150 DATA 16;CHAMPAGNE FRAISE;4;.1;24;.1;12;4;25;1;60
8155 DATA "POUR STRAWBERRY LIQUEUR AND KIRSCH INTO PRECHILLE
D CHAMPAGNE GLASS. TILT GLASS SO THAT LIQUEURS COAT BOTTOM A
ND SIDES OF GLASS. ADD CHAMPAGNE. FLOAT STRAWBERRY ON DRINK.
";END
8160 DATA 17;CHAMPAGNE COCKTAIL;4;.5;47;1;46;4;25;1;58
8165 DATA "STIR SUGAR ANF BITTERS IN PRECHILLED CHAMPAGNE GL
ASS. ADD CHAMPAGNE. USUALLY, THE SPARKLE OF THE CHAMPAGNE WI
LL BLEND THE INGREDIENTS, AND LITTLE STIRRING IS NECCESARY.
TWIST LEMON PEEL ABOVE DRINK AND DROP INTO GLASS.";END
8170 DATA 18;BRONX;4;1.5;32;.5;42;.25;15;.25;14
8175 DATA "SHAKE WELL WITH ICE. STRAIN INTO PRECHILLED COCKT
AIL GLASS. FOR A DRIER BRONX, OMIT SWEET VERMOUTH AND INCREA
SE GIN TO 1.75 OZS.";END
8180 DATA 19;FOGGY DAY;4;1.5;32;.25;23;1;57;1;58
8185 DATA "SHAKE GIN AND PERNOD WELL WITH ICE. STRAIN INTO P
RECHILLED OLD-FASHIONED GLASS. ADD ICE TO FILL GLASS. RUB OU
TSIDE OF LEMON PEEL AROUND RIM OF GLASS AND DROP PEEL INTO G
LASS. ADD LEMON SLICE.";END
8190 DATA 20;GIMLET;2;2;32;.5;44
8195 DATA "STIR EXTREMELY WELL WITH ICE. STRAIN INTO PRECHIL
LED COCKTAIL GLASS. LONG STIRRING IS ABSOLUTELY ESSENTIAL. G
LASS MAY BE SUGAR-FROSTED BY MOISTENING RIM WITH LIME JUICE
BEFORE DIPPING INTO SUGAR.";END
8200 DATA 21;MARTINI;2;2;32;.25;15
8205 DATA"A MARTINI MUST BE PIERCINGLY COLD; AT ITS BEST, BO
TH GIN AND VERMOUTH ARE PRECHILLED IN THE REFRIGERATOR, WELL
 STIRRED WITH ICE AND POURED INTO A PRECHILLED GLASS. ENERGE
TIC STIRRING WITH THE ICE IS ALL-IMPORTANT.";END
8210 DATA 22;FROZEN ORANGE BLOSSOM;7;1.5;32;2;42;.5;1;.5;38;
.1;49;4;41;.5;55
8215 DATA "PUT GIN, ORANGE JUICE, CURACAO, LEMON JUICE, ORAN
GE-FLOWER WATER AND ICE INTO BLENDER. SPIN 5-8 SECONDS. POUR
 INTO CHAMPAGNE OR OLD-FASHIONED GLASS. PLACE ORANGE SLICE O
N  TOP.";END
8220 DATA 23;PINK LADY;5;1.5;32;.25;44;1;39;1;43;.5;40
8225 DATA "SHAKE WELL WITH ICE. STRAIN INTO PRECHILLED COCKT
AIL GLASS. GLASS MAY BE SUGAR-FROSTED BY MOISTENING RIM WITH
 GRENADINE BEFORE DIPPING INTO SUGAR.";END
8230 DATA 24;BACARDI;3;1.5;30;.5;44;1;43;"SHAKE WELL WITH IC
E. STRAIN INTO PRECHILLED COCKTAIL GLASS OR OVER ROCKS IN A
PRECHILLED OLD-FASHIONED GLASS.";END
8240 DATA 25;CHERRY DAIQUIRI;5;1.5;30;.5;44;.5;6;.1;12;1;56
8245 DATA "SHAKE RUM, LIME JUICE, CHERRY LIQUEUR AND KIRSCH
WELL WITH ICE. STRAIN INTO PRECHILLED COCKTAIL GLASS. TWIST
LIME PEEL ABOVE DRINK AND DROP INTO GLASS.";END
8250 DATA 26;DAIQUIRI;3;2;30;.5;38;.5;47
8255 DATA "SHAKE WELL WITH ICE. POUR INTO PRECHILLED SUGAR-F
ROSTED COCKTAIL GLASS OR OVER THE ROCKS IN AN OLD-FASHIONED
GLASS. SUGAR MAY BE INCREASED FOR A SWEETER DAIQUIRI.";END
8260 DATA 27;FROZEN APPLE DAIQUIRI;6;1.5;30;.5;50;.5;38;.33;
41;1;47;1;55
8265 DATA "PUT RUM, APPLE JUICE, LEMON JUICE, CRUSHED ICE AN
D SUGAR INTO BLENDER. BLEND 10-15 SECONDS  AT LOW SPEED. POU
R INTO PRECHILLED CHAMPAGNE GLASS. ADD APPLE WEDGE.";END
8270 DATA 28;MAI TAI;8;3;30;.5;44;.25;3;.25;26;.5;47;1;56;1;
62;1;61
8275 DATA "SHAKE RUM;LIME JUICE;TRIPLE SEC;ORZATA AND SUGAR
WELL WITH ICE.STRAIN INTO PRECHILLED DOUBLE OLD-FASHIONED GL
ASS. ADD ENOUGH ICE TO FILL GLASS.GARNISH WITH LIME SLICE;MI
NT SPRIG AND PINEAPPLE STICK.";END
8280 DATA 29;SCORPION;7;2;30;2;42;1.5;38;1;2;.5;26;.33;41;1;
55
8285 DATA "PUT RUM;ORANGE JUICE;LEMON JUICE;BRANDY;ORZATA AN
D ICE INTO BLENDER. BLEND AT LOW SPEED 10-15 SECONDS. POUR I
NTO PRECHILLED DOUBLE OLD-FASHIONED GLASS WITH ENOUGH ICE CU
BES TO FILL GLASS TO RIM. ADD ORANGE SLICE.";END
```

```
8290 DATA 30,MARGERITA,3,1.5,37,.5,3,.5,38
8295 DATA "SHAKE WELL WITH ICE. STRAIN INTO PRECHILLED SALT-
RIMMED COCKTAIL GLASS. TO PREPARE GLASS, RUB RIM WITH OUTSID
E OF LEMON PEEL; THEN DIP INTO SALT AND SHAKE OFF EXCESS.",E
ND
8300 DATA 31,TEQUILA SOUR,4,2,37,.5,38,1,47,.5,57
8305 DATA "SKAKE TEQUILA, LEMON JUICE, AND SUGAR WELL WITH I
CE. STRAIN INTO PRECHILLED WHISKEY-SOUR GLASS. ADD LEMON SLI
CE.",END
8310 DATA 32,BLACK RUSSIAN,2,1.5,31,.75,27,"SHAKE WELL WITH
ICE. STRAIN OVER ROCKS IN PRECHILLED OLD-FASHIONED GLASS.",E
ND
8320 DATA 33,BLOODY MARY,6,1.5,31,3,51,.5,38,1,54,1,52,1,53
8325 DATA "SHAKE ALL INGREDIENTS WELL WITH ICE. STRAIN INTO
TALL OR SQUAT 8-OZ. GLASS.",END
8330 DATA 34,SCREWDRIVER,3,1.5,31,4.5,42,1,38
8335 DATA "SHAKE EXTREMELY WELL WITH ICE OR POUR INTO BLENDE
R AND BLEND WITH ICE AT HIGH SPEED FOR 5 SECONDS. STRAIN INT
O PRECHILLED TALL OR SQUAT 10-OZ. GLASS.",END
8340 DATA 35,SALTY DOG,3,2,31,.5,48,1,38,"SHAKE VODKA, GRAPE
FRUIT JUICE AND LEMON JUICE WELL WITH ICE. STRAIN INTO PRECH
ILLED COCKTAIL GLASS. SPRINKLE DISH WITH SEVERAL GENEROUS DA
SHES OF SALT.",END
8350 DATA 36,MANHATTAN,3,2,34,.5,14,1,46,"STIR WHISKEY, VERM
OUTH AND BITTERS WELL WITH ICE. STRAIN INTO PRECHILLED COCKT
AIL GLASS.",END
8360 DATA 37,OLD FASHIONED,5,.5,47,2,46,2,45,2,34,1,58
8365 DATA "STIR SUGAR, BITTERS AND SODA WATER IN PRECHILLED
OLD-FASHIONED GLASS UNTIL SUGAR DISSOLVES. FILL GLASS WITH I
CE CUBES OR LARGE PIECES OF CRACKED ICE. ADD WHISKEY. STIR W
ELL. TWIST LEMON PEEL ABOVE DRINK AND DROP INTO GLASS.",END
8370 DATA 38,RUSTY NAIL,2,.75,35,.75,21,"POUR OVER ROCKS IN
PRECHILLED OLD-FASHIONED GLASS. STIR.",END
8380 DATA 39,"TOM COLLINS",6,2,32,1,47,1,38,4,45,1,57,1,55
8385 DATA "SHAKE GIN, SUGAR, AND LEMON JUICE WELL WITH ICE.
STRAIN INTO TALL 14-OZ. GLASS HALF-FILLED WITH ICE. ADD SODA
. STIR. ADD LEMON SLICE AND ORANGE SLICE.",END
8390 DATA 40,BYRRH CASSIS COOLER,4,2,28,.5,10,4,45,1,57
8395 DATA "PUT ICE CUBES UP TO THE RIM IN A TALL 14-OZ. GLAS
S. ADD BYRRH AND CREME DE CASSIS. ADD SODA. STIR. GARNISH WI
TH LEMON SLICE.",END
8400 DATA 41,MINT JULEP,4,18,63,1,47,2,45,2.5,36
8405 DATA "TEAR 18 MINT LEAVES PARTIALLY WHILE LEAVING THEM
ON STEM, PLACE IN TALL 12-OZ. GLASS OR SILVER JULEP MUG WITH
 SUGAR AND CLUB SODA. STIR UNTIL SUGAR IS COMPLETELY DISSOLV
ED. FILL GLASS WITH FINELY CRACKED ICE. ADD BOURBON.STIR.",E
ND
```

FIGURE 5



*GO TO POINT MARKED * ON FIGURE 1

```
8410 DATA 42,GIN FIZZ,5,2,32,.5,38,1,47,4,45,1,57
8415 DATA "SHAKE GIN,LEMON JUICE AND SUGAR WELL WITH ICE. ST
RAIN INTO TALL 14-OZ. GLASS HALF-FILLED WITH ICE. FILL GLASS
 WITH SODA. STIR. ADD LEMON SLICE.",END
9000 REM *** HIGHBALL OR COLLINS GLASS
9010 PRINTCHR$(16);CHR$(22);:PRINT:PRINT:PRINT
9020 PRINTTAB(12);"X     X"
9030 PRINTTAB(12);"X     X"
9040 PRINT"HIGHBALL";TAB(12);"X     X"
9050 PRINT"   OR";TAB(12);"X     X"
9060 PRINT" COLLINS";TAB(12);"X     X"
9070 PRINT"  8-11";TAB(12);"X     X"
9080 PRINT"   OZS.";TAB(12);"X     X"
9090 FORI=1 TO 4:PRINTTAB(12);"X     X":NEXTI
9100 PRINTTAB(13);"-----"
9110 INPUT A$:RETURN
9120 REM *** CORDIAL GLASS
9130 PRINTCHR$(16);CHR$(22);:PRINT:PRINT
9140 PRINTTAB(12);"CORDIAL"
9150 PRINTTAB(13);"1 OZ.":PRINT:PRINT
9160 FORI=1TO4:PRINTTAB(14);"X   X":NEXTI
9170 PRINTTAB(15);"X X"
9180 FORI=1TO3:PRINTTAB(16);"X":NEXTI
9190 PRINTTAB(14);"-----"
9195 INPUT A$:RETURN
9200 REM *** SHOT OR JIGGER
9210 PRINTCHR$(16);CHR$(22);:PRINT:PRINT:PRINT
9220 PRINTTAB(10);"SHOT OR JIGGER":PRINT
9230 PRINTTAB(13);"1.5 OZS.":PRINT:PRINT:PRINT
9240 PRINTTAB(11);"X      X":PRINTTAB(11);"X      X"
9250 PRINTTAB(12);"X    X"
9260 FORI=1TO3:PRINTTAB(13);"XXXX":NEXTI
9270 INPUT A$:RETURN
9300 REM *** DELMONICO OR SOUR
9310 PRINTCHR$(16);CHR$(22);:PRINT:PRINT
9320 PRINTTAB(11);"DELMONICO
9330 PRINTTAB(14);"OR"
9340 PRINTTAB(13);"SOUR"
9350 PRINTTAB(11);"4-7 OZS.":PRINT
9360 FORI=1 TO 5:PRINTTAB(13);"X    X":NEXTI
9365 PRINTTAB(14);"XXXX"
9370 INPUT A$:RETURN
9400 REM *** ALL-PURPOSE WINE GLASS
9405 PRINT"WINE GLASS"
9410 PRINTCHR$(16);CHR$(22);
9420 FORI=1 TO 8:PRINTTAB(12);"X      X":NEXTI
9430 PRINTTAB(13);"X    X"
9440 PRINTTAB(14);"X  X"
9450 FORI=1TO3:PRINTTAB(15);"XX":NEXTI
9460 PRINTTAB(12);"--------"
9470 INPUT A$:RETURN
9500 REM *** COCKTAIL GLASS
9510 PRINTCHR$(16);CHR$(22);TAB(12);"COCKTAIL"
9520 PRINTTAB(12);"3-5 OZS."
9530 FORI=1TO5:PRINTTAB(10);"X          X":NEXTI
9540 PRINTTAB(11);"X        X"
9550 PRINTTAB(12);"X      X"
9560 PRINTTAB(13);"XXXXXX"
9570 FORI=1TO4:PRINTTAB(15);"XX":NEXTI
9580 PRINTTAB(11);"----------"
9590 INPUT A$:RETURN
9600 REM *** DEEP SAUCER CHAMPAGNE
9610 PRINTCHR$(16);CHR$(22);:PRINT
9620 PRINTTAB(8);"X    DEEP      X"
9630 PRINTTAB(8);"X   SAUCER     X"
9640 PRINTTAB(8);"X  CHAMPAGNE   X"
9650 PRINTTAB(9);"X   6 OZS.   X"
9660 PRINTTAB(9);"X";TAB(22);"X"
9670 PRINTTAB(10);"X";TAB(21);"X"
9680 PRINTTAB(11);"X        X"
9690 PRINTTAB(13);"X    X"
9700 PRINTTAB(14);"XXXX"
9710 PRINTTAB(15);"XX":PRINTTAB(15);"XX"
9720 PRINTTAB(14);"XXXX"
9730 PRINTTAB(13);"XXXXXX"
9740 PRINTTAB(11);"----------"
```

```
9750 INPUT A$:RETURN
9800 REM *** OLD FASHIONED
9810 PRINTCHR$(16);CHR$(22);:PRINT
9820 PRINTTAB(10);"OLD FASHIONED"
9830 PRINTTAB(8);"X   6-10 OZS.  X"
9840 FORI=1TO2:PRINTTAB(8);"X";TAB(23);"X":NEXTI
9850 FORI=1 TO 3:PRINTTAB(9);"X";TAB(22);"X":NEXTI
9860 FORI=1TO3:PRINTTAB(10);"X";TAB(21);"X":NEXTI
9870 FORI=1 TO 3:PRINTTAB(11);"X";TAB(20);"X":NEXTI
9880 PRINTTAB(11);"----------"
9890 INPUT A$:RETURN
9900 REM *** COOLER
9910 PRINTCHR$(16);CHR$(22);
9920 PRINTTAB(11);"X";TAB(20);"X"
```

FIGURE 7

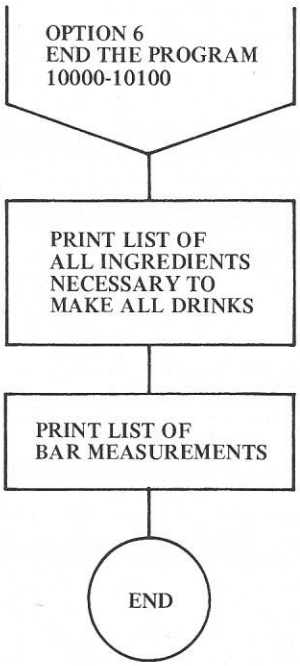
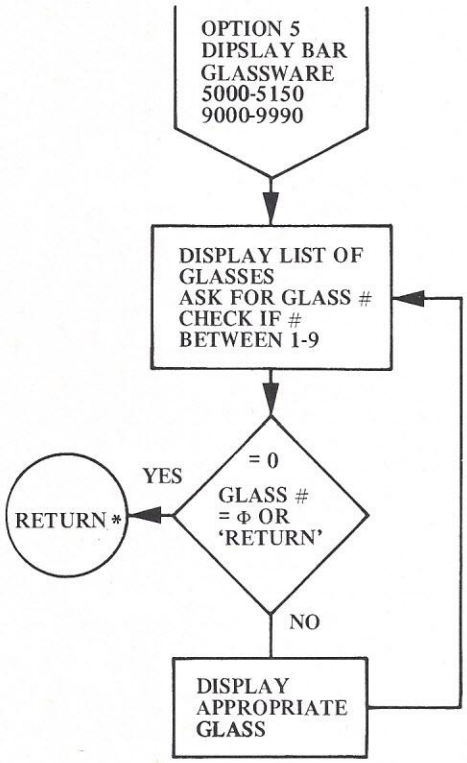
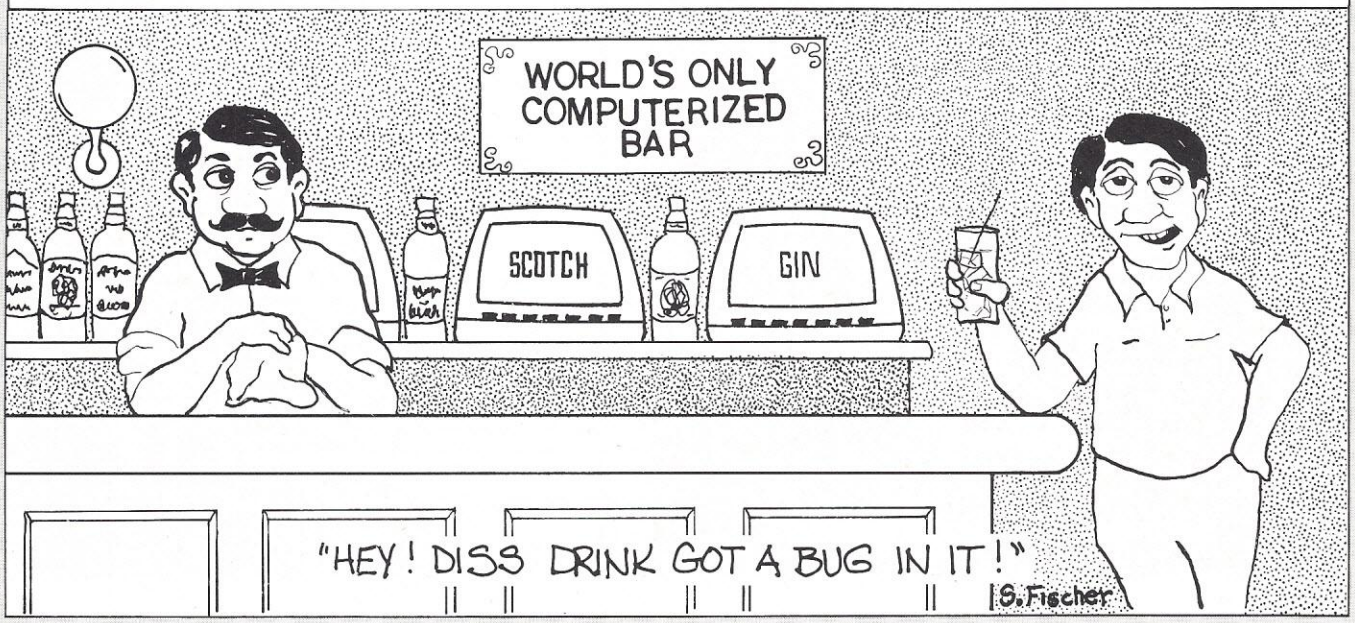


FIGURE 6



*GO TO POINT MARKED * ON FIGURE 1

```
9930 PRINTTAB(11);"X COOLER X"
9940 PRINTTAB(11);"X 14-21  X"
9950 PRINTTAB(11);"X  OZS.  X"
9960 FORI=1 TO 9:PRINTTAB(11);"X";TAB(20);"X":NEXTI
9970 PRINTTAB(12);"X";TAB(19);"X"
9980 PRINTTAB(13);"------"
9990 INPUT A$:RETURN
9999 DATA END
10000 REM *** FINAL OUTPUT ROUTINE
10010 PRINT:PRINTCHR$(16);CHR$(22);
10020 PRINT"TOTAL INGREDIENTS NEEDED"
10030 PRINT"TO MAKE THE DRINKS YOU SPECIFIED:"
10040 PRINT
10050 FOR I=1 TO 65
10060 IF IM(I)>0 THEN PRINTIM(I);TAB(7);:IN=I:GOSUB 3000:PRI
NT
10070 NEXT I
10080 PRINT:PRINT"END OF TABLE OF INGREDIENTS"
10090 PRINT:PRINT
10095 PRINT:GOSUB 4000
10100 END
```



Watts for Dinner

BY AUSTIN LESEA and RODNEY ZAKS

Power supply performance is measured by the following parameters: voltage and current ratings, regulation and efficiency.

Voltage, current rating

In review, power supplies are rated by the amount of energy or power they can deliver to the load by the voltage and current specifications.

For example, a five volt at five ampere power supply will deliver 25 watts to the load, maximum.

To know the voltages the circuits will require involves looking them up in the data sheets for your circuit. But how much current will the whole system draw? Again, the current requirements for each part are in the specification sheets. Listed as minimum, typical and maximum current drain, or alternately as power dissipation, one simply adds all the typical figures together to get a rough estimate of the operating current.

For specification of the power supply the current rating of the supply should be twice the average or typical system load current requirement. Maximum values should also be tabluated, as well as minimum values for power supply regulation specification.

Regulation

Power supplies are not perfect. They cannot deliver exactly 5.000 volts under all load conditions. Thus, they are also rated as to the capability to *regulate,* or hold the output voltage constant.

Specifications are divided into *load regulation tolerance* and *load-no-load regulation* tolerance. Also important is the *turn-on overshoot* and *stability* under varying load conditions.

If the load is constant and our input line voltage from the wall socket is within whatever range of values the utility company will allow, the variation in output voltage over temperature and time is the *load regulation tolerance.*

Man does not live by bread alone; but computers feast solely on electricity. And the chef who turns raw current into edible electricity for your computer is your power supply.

If the load varies you need to know how much the voltage will change – the *no-load, load regulation* measurement, or the *step-load* measurement. As an example, if you take a 5 volts at 5 amperes supply and attach a 1 ohm load, the supply should deliver 5 volts at 5 amperes. If you measure the voltage under this condition, and then disconnect the load and measure again, you should see optimally *no change.* In reality, step-load tolerance of less than 0.5% is usual.

In addition to the tolerance, the overshoot and stability measurements are extremely important. *Overshoot* occurs if the supply, when turned on or off, goes above the rated output voltage. *Stability* measures oscillating power supply under varying load conditions.

If we are driving a standard TTL, the circuits will not tolerate an overshoot of more than 8 volts; otherwise, the circuits will be destroyed. Most commercial power supplies have little or no overshoot.

Efficiency

If a supply delivers 25 watts to a load, how many watts is the supply taking from the wall socket? This ratio of output power divided by input power is the efficiency. Typical power supplies are 40% efficient. That is, the 25 watt model will draw 62.5 watts from the power line. High efficiency regulators of the *switching* type will deliver efficiencies of about 90%, but they are more expensive than the typical linear regulator.

The S-100 bus power supply

We've examined the operation of a simple linear regulator power supply together with voltage-current relationships, transformer action, rectifier, filter and regulator sections. Let's consider now the common power supply system affecting the S-100 bus.

The S-100 bus specifies that +8.0, +16.0 and -16.0 volts supplies be made available. The supplies can be "brute force" or unregulated, as the regulation will be performed on each board. The only requirement of such a supply is that it have a low output impedance; or simpler, that it will not be "soft". Soft implies if the load changes, the resulting voltage change will be great. With such a soft supply, the regulators might become unstable under certain conditions.

The brute force supply can be a simple choice of transformer, diodes and capacitors. Unfortunately, the design choices required are hardly ever observed. Saying a "bigger capacitor is better" is like saying one does his finest work with a sledge hammer. Large capacitors imply that diodes can handle the initial charge current when power is first applied.

Now, the +8.0 volt supply must get to the board with as little resistance or inductance as possible along the way,

necessitating thick printed circuit traces and large wires. As a general rule, more than one edge connector pin should be used for the power supplies. Also, these power pins should be "guarded" from other bus signals by grounding pins. Power and bus signals, therefore, will never mix due to poorly inserted or misaligned boards. The S-100 bus does not observe this convention.

Before considering the regulation, there is one last point: *bleeder resistors.* These are resistors across filter capacitors to bleed off any charge after the supply is turned off. If this bleeding is not done, the capacitors will remain charged for a long time and damage may result because power is still being applied to a non-functioning circuit.

When the unregulated voltages arrive at the boards, the voltages must then be regulated. The unused voltage margin times the current drawn must be dissipated as heat by the regulators. Placing the regulators on the same board as the chips will create a hotter local environment. A typical RAM board will draw 1.5 amperes at 5 volts. The unused margin of 3 volts times 1.5 amperes is 4.5 watts. This 4.5 watts must go up as heat on the board somewhere. Proper regulator IC mounting and heat sinking must be observed. Most S-100 products observe these mounting rules.

Quite often, the regulation requirements are split up between two or more regulator ICs – okay as long as the tolerance of these ICs are close. If one is at 4.75 volts and the other is at 5.25 volts, circuits running off different supplies may still communicate, but the noise margin is reduced. Yes, and it can get worse. The heat problem also affects noise margins.

In short, before the designer has even started, the system is doomed to strange, intermittent behavior due to noise. The regulators are graded by tolerance, so the particular lot of regulators used by a company may affect their product performance.

How can these problems be solved?

1) Match regulators to ±2.5% or provide adjustment.

2) Observe heat sinking rules carefully.

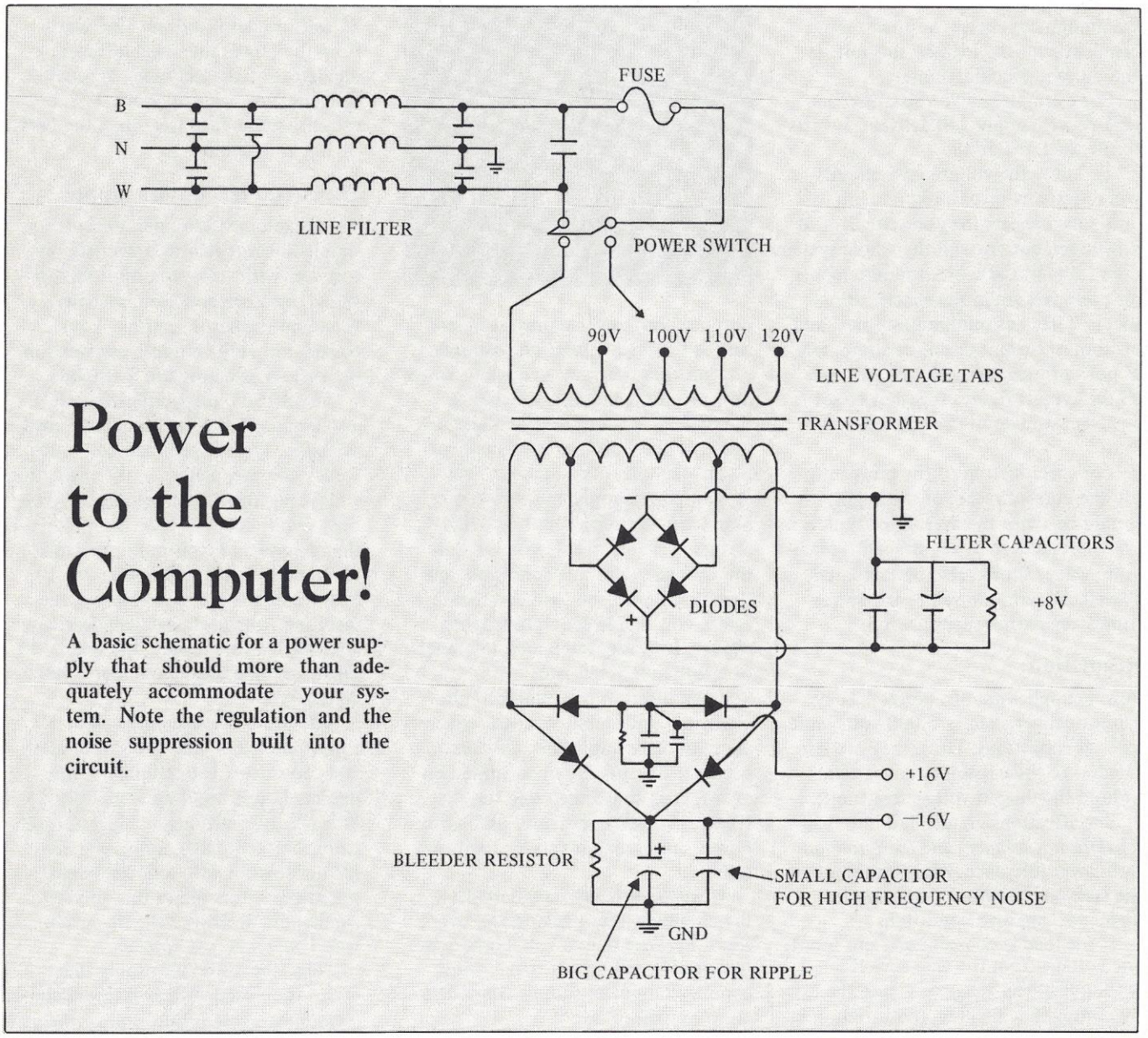
3) Follow bypassing rules religiously.

4) Use multiple traces or double wide PC runs for all power and ground runs.

5) Remember rule number 2. It should work at twice the power dissipated or current drawn. □

Power to the Computer!

A basic schematic for a power supply that should more than adequately accommodate your system. Note the regulation and the noise suppression built into the circuit.



Building a Benton Harbor Micro

BY BILL *AND KATHA* ENDRESS

Building a computer can involve the whole family. So when Bill Endress wrote this article, his wife, Katha, decided to add her view *(in italics)* of the experience.

Being a relative newcomer to the personal computing movement, I was bewildered at the tremendous variety of equipment from which to choose. To help narrow down my choice, I sent for catalogues and flyers on many different systems.

I also searched the various hobby magazines for product reviews and personal experience articles. Letters in the magazines dealing with individuals' problems and experiences getting one system or another up and running proved exceptionally helpful. I also visited several local computer shops.

One thing I knew from the start. You can't just go out and buy a lot of different equipment, hook it up and expect it to work the first time. I'd read many letters and talked to many people, and knew systems integration can be a problem.

Like many hobbyists, I waited eagerly for any scrap of news about Heathkit's new computer. When Heathkit finally announced the product in a well photographed, multi-page advertisement in all the popular personal computing magazines, I was delighted. *(It was all I could do to get that ad out of his hands at dinner time.)*

After examining Heath's literature, I was assured the system would function properly as a unit once assembled. All I had to do was build it, plug it in, flip a switch, and I'd be up and computing!

I decided to order the operating and reference manual set for the H8 system first. I recommend this purchase for any system you may seriously be considering. The expense involved in finding out the true capabilities of the system are negligible compared to the expense *(and possible heartbreak)* of purchasing a system which won't fill your needs.

After receiving the manual set, I found another good reason for ordering

Illustration by Casserine Toussaint

it is to digest it before your computer arrives! The software reference manual and the H8 operating manuals together equal over 450 pages.

The manual set consists of assembly and operating manuals for the H8 computer, the H9 video terminal, the H8-1 4K memory and 4K expansion set, the H8-5 I/O serial card, and the H10 paper tape punch and reader. The set also includes a software reference manual and a handsome, white, padded binder.

In the tradition of all Heathkit manuals, these were excellent. Anyone who has ever put together a Heathkit knows the quality of their assembly manuals. The operations manuals were every bit as complete.

The software reference manual and the H8 operating manual are invaluable aids. They do not, however, teach you programming from the ground up. If you're still a novice at computer programming, I suggest you purchase one of Heathkit's programming training courses.

After pouring over the manuals, I decided to take the plunge and order the Heathkit System One. This system includes their H8 8080A based computer, the H9 video terminal, 8K of memory, a serial I/O board and a cassette tape recorder. All I had to do now was shuffle the bank accounts to prepare for the $1200 expense and convince my wife of the need and importance of owning our own personal computer. *(That took some smooth talking. The part that convinced me was the size of the unit. It wouldn't take over the whole house like a UNIVAC.)*

Like any system, the Heathkit System One had some pros and cons. On the negative side was the bus structure. The Heathkit H8 doesn't use the S-100 bus system, so the system won't accept such popular items as color graphics, speech synthesizers and the multitude of other S-100 adaptable products now on the market.

But I didn't believe this would remain a problem. Heath would probably rush to offer these items as a part of their line, and other manufacturers would, in turn, also start making Heath-compatible products. Already there's a 12K memory board made especially for the H8 by Godbaud for a very reasonable price.

On the positive side, I felt Heath wasn't a fly-by-night company who'd disappear once the sale was made. They're a well established company with many years of success behind them. They have the financial ability to weather any storm in the personal computing field. And they also have a reputation for producing quality kits backed by a good service department.

At last I accumulated the money to buy the system and my wife was on my side – actually, I think she gave in out of sheer exasperation; she was probably tired of seeing me at breakfast, lunch and dinner with my nose in an assembly or operating manual, or hearing me praise the Heath system. *(True, but I really gave in because I thought it might be fun to have our own computer.)*

After what seemed like a year's wait *(it was only three weeks, and part of that was taken up by the Thanksgiving holiday)*, my wife called me at the office to say the boxes from Heathkit had arrived.

Practically any night of the year I can leave work a little early, but that night I had to work late! I broke out in a cold sweat waiting to get home and heat the old soldering iron.

When I finally arrived home, I could

Heathkit H9 CRT Terminal

hardly believe my eyes. Even though I'd ordered a complete system, I wasn't prepared for the number of boxes before my eyes – six boxes, and that total didn't even include the video terminal which was being shipped separately. Better than Christmas any old day!

After clearing the dining room table, *(it wasn't seen again for five days, much to my dismay)* and taking out the necessary tools, I started construction.

The construction of the computer itself is very straightforward. Just be sure, in your haste to get your system up and running, you don't forget to note any changes to the manual. Most new kits usually contain several errata sheets.

Heath's CPU board comes already assembled and tested, which eliminates much of the guesswork. If your system does not function properly the first time you apply power, you can probably eliminate the CPU as the cause of your troubles.

Heathkit quality is quite good. All of the ICs are socketed and the individual circuit boards firmly mounted to the chassis with nuts and bolts. Changing boards quickly can be a pain, but fortunately these changes are not made often. The firm mounting does assure you the boards will not be accidentally knocked out of position once you install them.

Both the 8K memory board and the serial I/O board also assemble smoothly.

Heath added an extremely handy extra circuit on the I/O board – an on-board logic probe. This probe is first used to check the logic circuits on the board. It is then wired to monitor incoming data from the cassette player. An on-board logic probe is a terrific idea. Too bad they don't put one on all the boards. Are you reading this, Heathkit?

After running the initial tests and installing the boards, I threw the switch. Amazing! No puff of tell-tale smoke, no ominous pops and no local brownouts. I entered the initial test routine according to the procedures in the operating manual and pushed the 'GO' button.

At this point Heathkit has to be commended. For lo and behold, there, flashing on the LED displays were the words: "your H8 is up and running" followed by three goal posts and two beeps on the speaker. This message repeats until the reset button is pushed.

I could hardly believe my eyes. Here was real, tangible evidence that the fruits of my labors had blossomed. Here was something I could show my wife! *(He actually woke me up from a sound sleep to come see this display. After my eyes focused, it was . . . well . . . the computer did spell all the words correctly.)*

After the memory checkout procedures were done, it was off to the races! I plugged in the tape recorder, slipped in the software cassette and punched the "LOAD" button. Nothing happened. According to the manual, the displays are supposed to flash the addresses of the incoming data. And if the load is successful, it'll be followed by a beep at the end.

Since the manual also stated it might be necessary to vary volume and tone on the recorder to achieve a successful load, I rewound the tape and pressed the "LOAD" button again. This time I varied the volume and tone controls through their full range. Again, nothing happened.

I removed the board from the com-

puter and gave it a thorough going-over. After checking the x-ray foil diagrams and going step by step back through the assembly manual, I reinserted the board and checked the adjustments.

Finding nothing wrong, I hooked the recorder back up to the system and pressed the "LOAD" button. As before, nothing happened.

In exasperation, I decided to listen to the tape. I pressed the "PLAYBACK" button on the recorder. No sound came from the tape. Never having heard a computer recording, I wasn't sure what I was supposed to hear. Perhaps I couldn't hear anything because the sounds were above the threshold of human hearing.

I sat there trying to perceive these ultrasonic sounds with the volume wide open. After about three minutes, just as I thought I heard some sounds, my ear drums were almost shattered by a god-awful, high pitched shriek. By the time I recovered my senses and turned down the volume, the shriek was replaced by noises that sounded like a portion of the sound track from "Star Wars". *(I thought he must have stepped on the cat).*

Hoping this noise might be the program, I rewound the tape to where I first heard the shriek. I plugged the recorder back into the computer and punched the "LOAD" button. Suddenly the front panel started reeling off address locations. After about a minute of anxious waiting, the numbers stopped and I heard a long awaited, satisfying beep. The load was successful.

I now had a working system which, with the exception of the video terminal, was complete.

About a week after I finished building the computer, the video terminal arrived. *(My table was covered again.)*

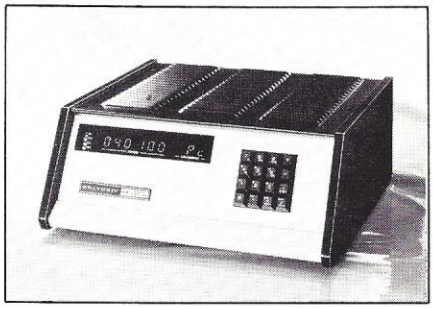
I've put together some kits in my time, including a stereo system and an oscilloscope, but none as complex as the video terminal. Fortunately, Heath did a superb job on the instruction manual so I was able to put the terminal together with only a few minor problems.

Construction is broken down into eight distinct steps – the chassis and seven circuit boards. Tests at the end of each step help detect any possible errors in construction.

When I was ready to begin the construction of the video terminal, my wife expressed an interest in helping with the soldering. *(I'm not sure if I was as interested in soldering as in getting my table back. It was nice, though, to be able to talk to him about the terminal and to understand which part he was talking about.)*

Since she had never constructed anything electronic before, I cautiously said "OK". I decided to let her start by constructing the power supply. The foils on the power supply board were, for the most part, large and had good separation. This feature helped prevent soldering bridges until she learned just how much solder should be applied. *(Solder certainly melts faster than I thought it would.)*

With the power supply's wide variety of components, my wife could also learn to distinguish the different parts. She did learn the same part could be packaged many different ways. *(I found out you have to look very closely to read the numbers on some parts. The power supply had ICs that looked like transistors, and capacitors that looked like resistors. Only a diode always looked like a diode.)*



Heathkit H8 Digital Computer

We started on the construction of the video terminal, she merrily soldering along *(and he checking my work very carefully).* Construction proceeded smoothly, with everything going as planned. In just a few days we had the terminal finished and checked out.

When you purchase the H8 computer you also receive a complete software package at no additional charge. The software includes PAM-8, BUG-8, TED-8, HASL-8 and BENTON HARBOR BASIC. For an additional charge of only $10 you can also get an extended version of their BENTON HARBOR BASIC.

PAM-8 (Panel Monitor) is the operating system that resides permanently in 1K of ROM. It controls the front panel of the computer as well as loading and dumping routines for the cassette tape system.

The front panel on the H8 is really super! If you have seen pictures of the H8, you probably have noticed a lack of familiar switches and status LEDs. Instead it has a pad of push buttons resembling the keys of an adding machine, and nine seven-segment LED displays. *(I just can't imagine anyone having the patience to flip all those switches that are on so many of the other computers.)*

The front panel, in conjunction with PAM-8, gives you complete control over your system. You can examine the contents and change a memory location, an I/O port, any register including the accumulator, the stack pointer, the program counter and the flag registers. Because of this control, you can directly enter and execute a program in machine language. You can also turn the speaker on and off.

You can control the lighting of each segment of the seven-segment displays, which allows you to display alphanumeric data on the front panel using all of the numbers and a limited alphabet. The front panel is also used to perform the cassette dumping and loading operations, master reset and program execution.

Another feature of PAM-8 is the generation of a real-time clock called TICCNT (Tick Counter). The clock is contained in two consecutive memory locations forming a 16-bit counter updated every 2 ms.

You use BUG-8 (Console Debugger) to enter and debug machine language programs from a video terminal – which is very handy when you enter a long program.

In BUG-8, machine codes can appear as different forms. Data can be entered using offset octal, decimal, or ASCII characters. So suppose you want to set up a program in machine language to perform a series of calculations and output the answer along with a message. You can enter the program body in octal form, the constants in decimal form and the message in ASCII form. You don't have to convert decimal or ASCII characters into octal. This feature is ideal for writing your own operating system for a video terminal, cassette record handler or any other special device.

Heath's software package also includes an assembler (HASL-8: Heath Assembly Language) for writing programs in assembly language and TED-8 (Heath Text Editor). I must confess I've had little opportunity to use these two pieces of software, so I'll pass on commenting.

Heath's special variety of BASIC is called, naturally enough, BENTON HARBOR BASIC. This BASIC, as well as all of the software put out by Heath, is a product of Wintek Corpo-

ration and is designed especially for the Heath H8.

With one exception, BENTON HARBOR BASIC seems to be a fairly complete BASIC. It has no string handling capabilities, except for the PRINT and INPUT statements. To obtain string handling capabilities you must move up to 12K of memory and purchase EXTENDED BASIC.

The BASIC has two modes of operation, a command mode and a program mode. The command mode is used for listing and running programs. It can also be used to turn your video terminal into a super calculator. For instance, by typing in the following: PRINT 10*5/6+3 the solution will print out when the return button is hit. You activate the program mode by prefixing all statements with numbers.

Both the command mode and program mode use command completion – an item I definitely don't like. Here is the way it works. As soon as "enough" letters are typed on the screen, for the computer to recognize this word as a "unique" command word, the computer automatically completes the remainder of the word and adds the necessary space or character at the end of the word.

For example, suppose you wanted to type the command PRINT. As soon as you type the letters PR, the computer identifies this as uniquely belonging to the word PRINT and adds the letters INT to the end of PR. In the meantime, the uninitiated (who are unaware of this feature) continue typing the letters INT. When the computer sees INT it identifies the command with the function of returning an integer value of an expression which is put into the computer in this form: INT(X). (The computer always adds the open parenthesis symbol).

So, typing a simple command PRINT results in the following: PRINT INT. Frustrating to say the least! *(Command completion would be much simpler to learn if the codes to be typed into the computer were listed somewhere in the manual. As it is now, you must check the screen after you type each letter until you memorize just how much of each command you actually type and how much the computer completes.)*

To compound the problem of command completion, I did not catch the explanation on backspacing the first time I read the manual. In BASIC, the ASCII character that represents a true backspace is formed by pressing the control and H buttons simultaneously. Instead I used the "←" to position the cursor. Needless to say I spent many a trying moment wrestling with command completion.

Another problem with BENTON HARBOR BASIC – one I suspect is common to all smaller versions of BASIC – is that the output numbers do not come out rounded off correctly. It wasn't until later I found out the problem is probably in the conversion from binary to decimal in the BASIC interpreter itself.

I was rounding off my interest calculation first, performing a series of additions and subtractions and then printing the final results. Apparently, a small error in the binary to decimal conversion was compounding itself with every calculation.

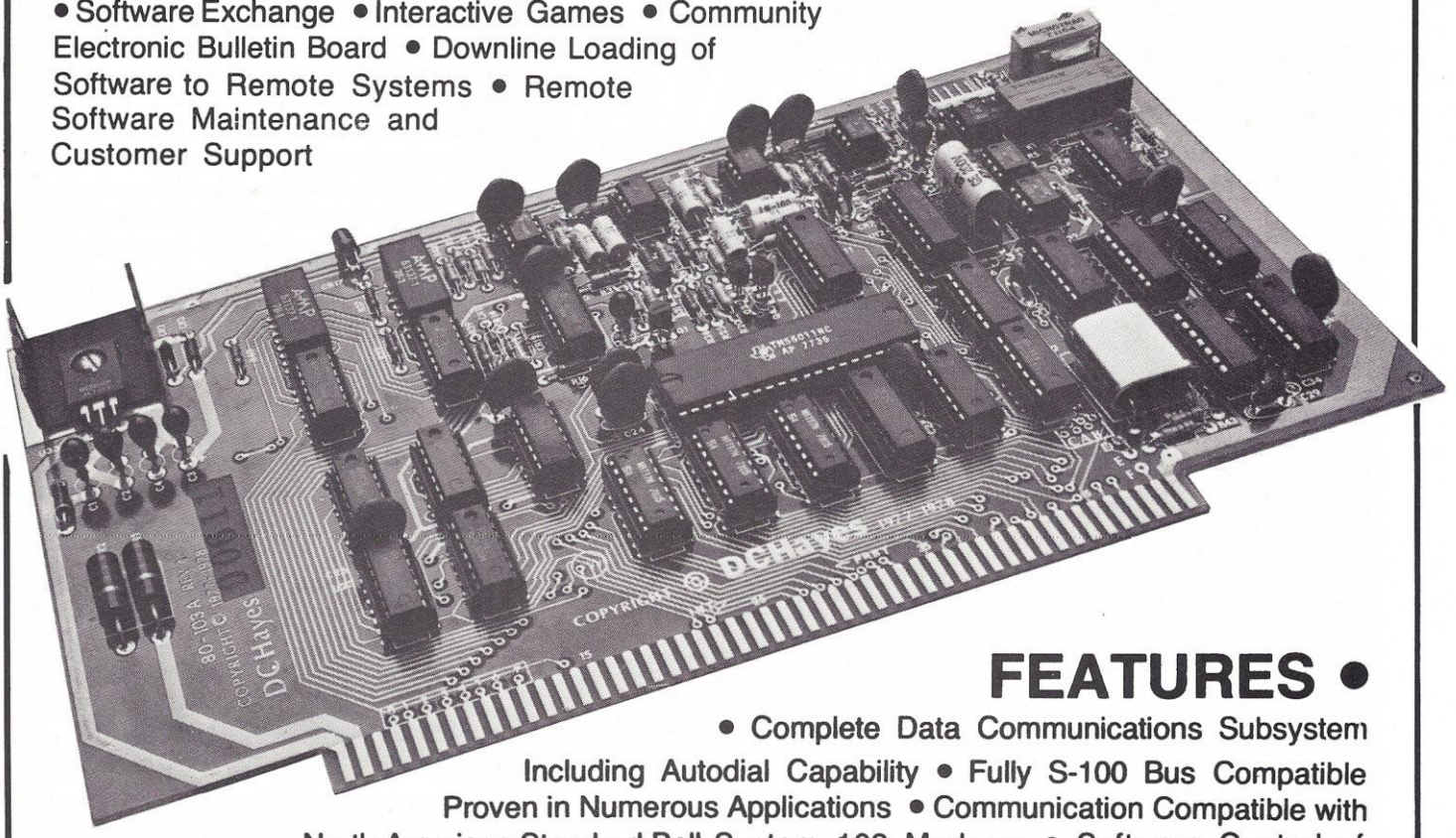
I was able to work around the problem by doing the addition and subtraction first, and then rounding off.

So there you have it. An in-depth look at the Heathkit H8 System One from a guy *(and a gal)* who own one. Do we like it? You bet we do. □

modem / 'mo • dəm / [**mod**ulator + **dem**odulator] *n* - *s* : a device for transmission of digital information via an analog channel such as a telephone circuit.

USES •

• Intelligent Terminal • Timesharing • Distributed Processing • Automatic Data Collection • Access to Remote Data Base • Line Concentrator • Telecommuting • Electronic Mail • Remote Access to Process Control Systems • Software Exchange • Interactive Games • Community Electronic Bulletin Board • Downline Loading of Software to Remote Systems • Remote Software Maintenance and Customer Support



FEATURES •

• Complete Data Communications Subsystem Including Autodial Capability • Fully S-100 Bus Compatible Proven in Numerous Applications • Communication Compatible with North American Standard Bell System 103 Modems • Software Control of Originate and Answer Format, and Parity • Extensive Self-Test Capability • All Digital Modulation and Demodulation Means No User Adjustments Required • On-Board Crystal Oscillator Insures Proper Timing in all S-100 Systems • Break Generation • Meets or Exceeds all Bell System and FCC Specifications for use with a CBT Coupler (Data Access Arrangement -DAA) • 90 Day Limited Warranty

D.C. Hayes Associates, Inc.

16 PERIMETER PARK DR. SUITE 101

P.O. BOX 9884 ATLANTA, GEORGIA, 30319 (404) 455-7663

DISTRIBUTED IN CANADA BY TRINTRONICS, LTD., TORONTO

CIRCLE 14

KEEPING TABS ON CONTRACT LABOR COSTS

BY KAREN S. WOLFE

Keeping track of your contract labor costs saves you money.

Knowing how much you spent in labor for each job lets you compare estimated labor costs (used to make bids) to the actual costs incurred. This comparison will tell you whether you need to change your bidding or labor processes.

If you run a plumbing, air conditioning, painting or electrical contracting firm (or any firm working on a contractual basis), keeping track of contract labor costs will also save you money when figuring state and city sales taxes.

This program, CONLAB, provides a simple way for you to keep tabs on your costs. Just input your labor costs, straight from your workers' time cards, and the computer will store those costs by job (contract) number and by employee (or department) number.

Labor costs can be deducted from total job price when figuring state and city sales taxes. (Note: Since regulations vary from state to state and city to city, you should check local regulations to find out exactly what is deductable in your area.)

For example, suppose an air conditioning firm contracts to install a residential air conditioner for $1500. If their labor costs are $800, they can deduct this amount from their total job price. Thus, their taxable revenue equals $1500 - $800, or $700. So if their local sales tax rate is 0.4%, their sales tax would come to $700 x 0.04, or $28.

CONLAB was designed for small subcontractors with 20 or fewer employees. The file holds up to 10 job numbers at any one time. However, these employee and job number constraints can be enlarged by making the J(I,C) data file matrix larger, where I = number of job numbers and C = number of employees + 4.

The program was written in North Star BASIC and designed for North Star disk storage. Multiple line statements are separated by "/". If you're using a different BASIC version, your system may require a change in CONLAB statement lines 10-40 and 280-310 for accessing the data file and resaving the file.

Initializer Program

First, you must create, or reserve, space on the disk for the data file, which I call LABOR, by following your system's instructions. Then load BASIC and use the Initializer Program (see Program Listing 1) to structure the data file and

fill all matrix cells with zeros. Again, your system may require a change in lines 35, 50 and 65 of the Initalizer Program to open, create and save the structured file. *Use this program only when you want all matrix cells to equal zero.*

CONLAB Program

Table 1 shows the data file matrix configuration. Each row contains information for a specific job with the actual job or contract number located in column 21. Columns 1 through 20 correspond to employee numbers (i.e., employee number 1, employee number 2, etc.).

Overhead Factor

Column 22 accumulates the direct labor costs of all employees. Column 23 is the product of column 22 times a labor overhead factor. This factor is a combination of indirect labor expenses. For example:

Soc. Sec. Employer Cont.	6.05%
State Unemployment	2.70
Federal Unemployment	.50
Workmen's Compensation	10.00
Total	19.25%

You'll probably have to estimate your workmen's compensation factor since different employees fall into varying tax rates depending on the nature of their work.

In the example above, the total factor is 19.25% of direct labor. Add this percentage to direct labor costs to generate actual labor expense. When you enter this factor in the program, however, it should be in decimal form – in this case, 0.1925.

The First Run

The first time you run CONLAB, you'll need to put in job numbers. After the program lists the currently stored job numbers and their total costs to date, it allows you three options: update job numbers, update labor costs or receive a report. On your first run, enter "1" to update job numbers. This command produces a listing of the rows and job numbers.

Of course, on the first pass all job numbers are zero. You're asked to enter the row number whose job number you wish to change. Next, you're asked to enter the new job number. Follow this procedure to fill the matrix with contract numbers. (See Sample Run.)

In the future, you'll need to add new contract numbers and delete others. Remember, when you input a row number, all the previous information in that row is made equal to zero. So be certain you're finished with that job's information before you input that row number and press return.

However, if you do accidentally delete the information in a row, you haven't completely lost the data. Suppose you mistakenly enter row 3 as the one you wish to replace with another job number, but you still need row 3's job information. At this point you've only deleted row 3's data in the computer memory, *not* the disk file. So clear the machine and re-load the program, which will call the unchanged data file off the disk. Row 3's data will still be intact.

Remember: *you have not changed the data on the disk until you tell the computer to save the updated file!* Statement 270 asks, "Do you wish to save this file now? (y/n)." If you enter "y", the data file you've been working on will be saved and the old data eliminated. But if you enter "n", the file on the disk is left unchanged, no matter what you've done to the data in the computer memory.

Sample Run

Assume you have data in the file from past pay periods and are going to add more labor from this week's time cards. First, a list of the job numbers and total costs to date are shown. Then you're asked to choose an option. You enter "1" because you have a new job number to enter.

A new listing shows row numbers and the job numbers they contain. Job number 7820 is completed and you no longer need the information from it, so you enter row "2" to be changed and enter "7827" for a new job. You're asked if you want to update more rows, and you enter "n".

The next question is, "Do you wish to update labor costs now? (y/n)." You enter "y". Then you're asked to input the overhead factor (for example, 0.1925).

CONLAB lets you take an employee's time card and enter all the jobs he worked on before you go on to the next employee.

Suppose you have only two employees, numbers 3 and 5, working on contract jobs for that pay period. Statement 90 asks you to enter an employee number, so you respond with "3". Then input his hourly rate of pay, $8.50. Next, enter a job number he worked on, "7827". To post labor for an entire week, add the number of regular hours employee number 3 worked on job 7827 and enter that total for the prompt, "Enter regular hours. . ." In this case, enter "32" hours. Did he work overtime on the job? For the prompt, "Enter overtime hours. . .", input "1" hour.

Next, you're asked for another job number. You enter "7816", "8" regular hours and 0 overtime hours. This employee worked on no other jobs, so when asked again to input another job number for this employee, you enter zero, "0". You're asked if there are more employees to enter; you respond with "y".

You'll then be asked for a new employee number; repeat the above routine. After the employee is entered and you're asked if there are more employees to enter, you input "n", causing the accumulation calculations to be made.

Then, a listing of job numbers and total labor costs to date, including labor overhead, is presented. Statement line 255 uses North Star's formatting procedure (%6I and $8F2). Your BASIC may require a change here.

LABOR Data File Matrix

	Employee Numbers					Job No.	Direct Cost	Col.22 +Overhead	Extra Col.
Col. Nos.	1	2	3	4. . . .	20	21	22	23	24
Row Nos. 1	*	*	*	*	*	*	*	*	*
2	*	*	*	*	*	7820	*	*	*
•									
•									
•									
10	*	*	*	*	*	*	*	*	*

The stars (*) represent data cells. For instance, job number 7820 is in the 2nd row, 21st column, therefore, it is in the J(2,21) data cell.

If you run a firm working on a contractual basis, keeping track of contract labor costs will save you money when figuring state and city sales taxes.

Now, you can either copy the total costs of the jobs or have this listing printed out in hardcopy form. Lines 245 through 260 create the listing, so you could add a printer select statement before line 245 and a monitor select after line 260 (perhaps at lines 243 and 267 respectively).

If you want additional information listed, you can request the desired J (I,C) matrix cell to be printed (refer to table 1). The unused column, 24, is for your convenience in case you need to save some piece of information not already included.

Finally, the program asks if you want to save this updated file. You enter "y" and the file is saved.

If you don't want to update any information but only want a list of contract numbers and total costs, enter "3" when asked the option question. If you've added the printer select statement in line 243, you'll receive a hardcopy report.

I suggest you make and save hardcopy, especially if you're using CONLAB for sales tax purposes. In fact, you may want a listing of the entire matrix.

Also, find out which overhead factors are deductible for sales tax purposes in your state and the percentage rates your firm pays on those factors. Don't cut yourself short. □

Sample Run

```
READY
RUN

     7815              256.39
     7820              307.07
        0                 .00
        0                 .00
     7816              246.85
        0                 .00
        0                 .00
        0                 .00
        0                 .00
        0                 .00

1.UPDATE JOB NOS.---- 2.UPDATE LABOR COSTS---- 3.REPORT

ENTER 1,2 OR 3 1
 1     7815
 2     7820
 3     0
 4     0
 5     7816
 6     0
 7     0
 8     0
 9     0
 10     0

ENTER ROW NO, YOU WISH TO UPDATE 2

ENTER JOB NUMBER YOU WISH FOR THIS ROW 7827

DO YOU WISH TO UPDATE MORE ROWS? (Y/N) N
DO YOU WISH TO UPDATE LABOR COSTS NOW? (Y/N) Y

ENTER OVERHEAD FACTOR IN DECIMAL FORM .1925

ENTER EMPLOYEE NUMBER 3
ENTER HOURLY RATE OF PAY 8.50

ENTER 0 (ZERO) FOR JOB NO. TO GO TO NEXT EMPLOYEE

ENTER JOB NUMBER 7827
ENTER REGULAR HOURS ON JOB NO.  7827 ?32
ENTER OVERTIME HOURS ON THIS JOB 1
ENTER 0 (ZERO) FOR JOB NO. TO GO TO NEXT EMPLOYEE

ENTER JOB NUMBER 7816
ENTER REGULAR HOURS ON JOB NO.  7816 ?8
ENTER OVERTIME HOURS ON THIS JOB 0
ENTER 0 (ZERO) FOR JOB NO. TO GO TO NEXT EMPLOYEE

ENTER JOB NUMBER 0

ARE THERE MORE EMPLOYEES TO ENTER? (Y/N) Y
ENTER EMPLOYEE NUMBER 5
ENTER HOURLY RATE OF PAY 5

ENTER 0 (ZERO) FOR JOB NO. TO GO TO NEXT EMPLOYEE

ENTER JOB NUMBER 7815
ENTER REGULAR HOURS ON JOB NO.  7815 ?40
ENTER OVERTIME HOURS ON THIS JOB 0
ENTER 0 (ZERO) FOR JOB NO. TO GO TO NEXT EMPLOYEE

ENTER JOB NUMBER 0

ARE THERE MORE EMPLOYEES TO ENTER? (Y/N) N
```

```
********** CALCULATIONS **********

 JOB NO.     TOTAL LABOR COST
  7815        494.89
  7827        339.56
     0           .00
     0           .00
  7816        327.94
     0           .00
     0           .00
     0           .00
     0           .00
     0           .00

DO YOU WISH TO SAVE THIS FILE NOW? (Y/N) Y
READY
```

Program Listing 1 – Initializer Program

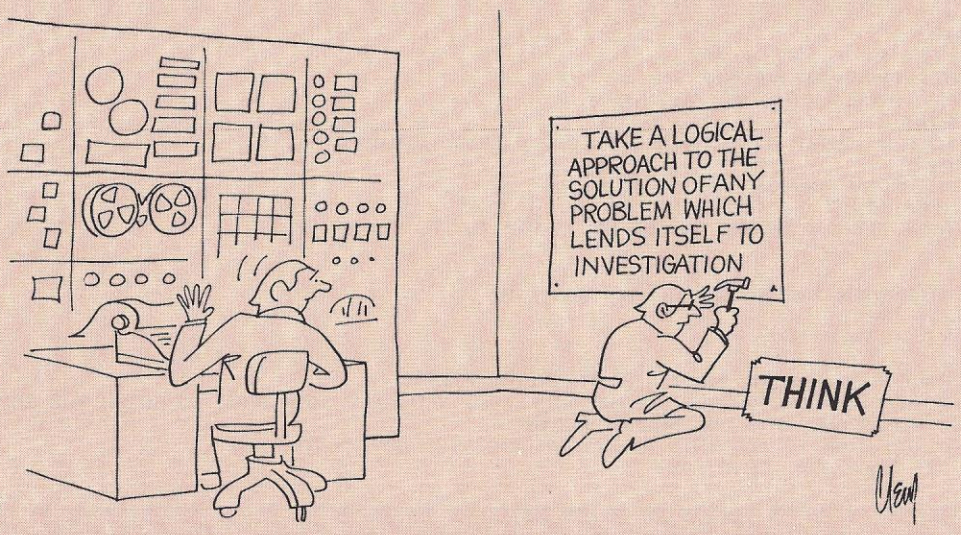
```
1 REM----INITIALIZER PROGRAM TO STRUCTURE DATA CELLS
5 DIM J(10,24)
10 FOR R= 1 TO 10
15 FOR C= 1 TO 24
20 LET J(R,C) = 0
25 NEXT C
30 NEXT R
35 OPEN #0, "LABOR"
40 FOR R= 1 TO 10
45 FOR C= 1 TO 24
50 WRITE #0, J(R,C)
55 NEXT C
60 NEXT R
65 CLOSE #0
70 END
READY
```

Program Listing 2 – CONLAB

```
1 REM----CONTRACT LABOR PROGRAM (CONLAB)
5 DIM J(20,24), D$(1)
10 OPEN #0, "LABOR"
15 FOR I= 1 TO 20
20 FOR C= 1 TO 24
25 READ #0, J(I,C)
30 NEXT C
35 NEXT I
40 CLOSE #0
45 FOR I= 1 TO 10
50 PRINT %8I, J(I,21), %16F2, J(I,23), %20I,J(I+10,21),%16F2,J(I+10,23)
55 NEXT I
60 PRINT
65 PRINT "1.UPDATE JOB NOS.---- 2.UPDATE LABOR COSTS---- 3.REPORT"
66 PRINT
70 INPUT "ENTER 1,2 OR 3 ",D2
75 IF D2=1 THEN 320
77 IF D2=3 THEN 243
80 PRINT\ PRINT\ PRINT\ PRINT\ PRINT
82 INPUT "ENTER OVERHEAD FACTOR IN DECIMAL FORM ",F
83 LET F= F + 1
84 PRINT\ PRINT
85 LET O= 0\ LET R= 0\ LET H= 0
90 INPUT "ENTER EMPLOYEE NUMBER ",C
95 INPUT "ENTER HOURLY RATE OF PAY ",R
100 PRINT\ PRINT
105 PRINT "ENTER 0 (ZERO) FOR JOB NO. TO GO TO NEXT EMPLOYEE"
110 PRINT\ PRINT
115 INPUT "ENTER JOB NUMBER ",B
116 IF B= 0 THEN 170
125 PRINT "ENTER REGULAR HOURS ON JOB NO. ",B," "
130 INPUT H
135 LET K=0
140 INPUT "ENTER OVERTIME HOURS ON THIS JOB ",O
145 FOR I= 1 TO 20
150 IF J(I,21)= B THEN 160
155 NEXT I
156 PRINT\ PRINT "JOB NUMBER ",B," IS NOT IN THE FILE"
157 PRINT "IF YOU  ADD THE NUMBER TO THE FILE NOW, YOU WILL LOOP BACK TO ENTER THE EMPLOYEE
      NUMBER AGAIN AND THIS JOB NUMBER AGAIN"
158 INPUT "DO YOU WISH TO ADD THIS NUMBER TO THE FILE NOW (Y/N) ",D$
159 IF D$ = "Y" THEN 320\ GOTO 105
```

Program Listing 2 – continued

```
160 LET J(I,C)= J(I,C) + (H*R) + (O*(1.5*R))
165 GOTO 105
170 PRINT\ PRINT\ PRINT\ PRINT
175 INPUT "ARE THERE MORE EMPLOYEES TO ENTER? (Y/N) ",D$
180 IF D$= "Y" THEN 85
185 PRINT\ PRINT\ PRINT\ PRINT
190 PRINT "        ********** CALCULATIONS **********"
195 PRINT\ PRINT\ PRINT\ PRINT
200 FOR I = 1 TO 20
205 LET J(I,22)= 0
210 NEXT I
215 FOR I= 1 TO 20
220 FOR C= 1 TO 20
225 LET J(I,22)= J(I,22) + J(I,C)
230 LET J(I,23)= J(I,22) * F
235 NEXT C
240 NEXT I
243 REM----PLACE PRINTER SELECT COMMAND HERE IF DESIRED
245 PRINT " JOB NO.  TOTAL LAB COST        JOB NO.  TOTAL LAB COST"
250 FOR I= 1 TO 10
255 PRINT %6I, J(I,21),"    ",%8F2, J(I,23),%20I,J(I+10,21),%11F2,J(I+10,23)
260 NEXT I
265 PRINT
267 IF D2= 3 THEN 400
270 INPUT "DO YOU WISH TO SAVE THIS FILE NOW? (Y/N) ",D$
275 IF D$= "N" THEN 400
280 OPEN #0, "LABOR"
285 FOR I= 1 TO 20
290 FOR C= 1 TO 24
295 WRITE #0, J(I,C)
300 NEXT C
305 NEXT I
310 CLOSE #0
315 GOTO 400
320 FOR I= 1 TO 10
325 PRINT %8I,I,%16I,J(I,21),%20I,I+10,%16I,J(I+10,21)
330 NEXT I
335 PRINT
340 INPUT "ENTER ROW NO, YOU WISH TO UPDATE ",I
345 FOR C= 1 TO 24
350 LET J(I,C)= 0
355 NEXT C
360 PRINT\ PRINT
365 INPUT "ENTER JOB NUMBER YOU WISH FOR THIS ROW ", J(I,21)
370 PRINT\ PRINT
375 INPUT "DO YOU WISH TO UPDATE MORE ROWS? (Y/N) ",D$
380 IF D$= "Y" THEN 320
385 INPUT "DO YOU WISH TO UPDATE LABOR COSTS NOW? (Y/N) ",D$
390 IF D$= "Y" THEN 80
395 GOTO 270
400 END
```



Radio Shack's personal computer system? This ad just might make you a believer.

You can't beat the 4K system at $599

TRS-80 "Breakthru"
- TRS-80 microcomputer
- 12" video display
- Professional keyboard
- Power supply
- Cassette tape recorder
- 4K RAM, Level-I BASIC
- 232-page manual
- 2 game cassettes

...or the step-up 16K system at $899

TRS-80 "Sweet 16"
- Above, except includes 16K RAM

...or the fast 4K/printer system at $1198

TRS-80 "Educator"
- Above, except includes 4K RAM and screen printer

...or the Level-II 16K/printer/disk system at $2385

TRS-80 "Professional"
- Above, except includes 16K RAM, disk drive, expansion interface, and Level-II BASIC

So how are you gonna beat the system that does this much for this little? No way!

...The amazing new 32K/Level-II/2-disk/ line printer system at $3874

TRS-80 "Business"
- Above, except includes 32K RAM, line printer, and two disk drives

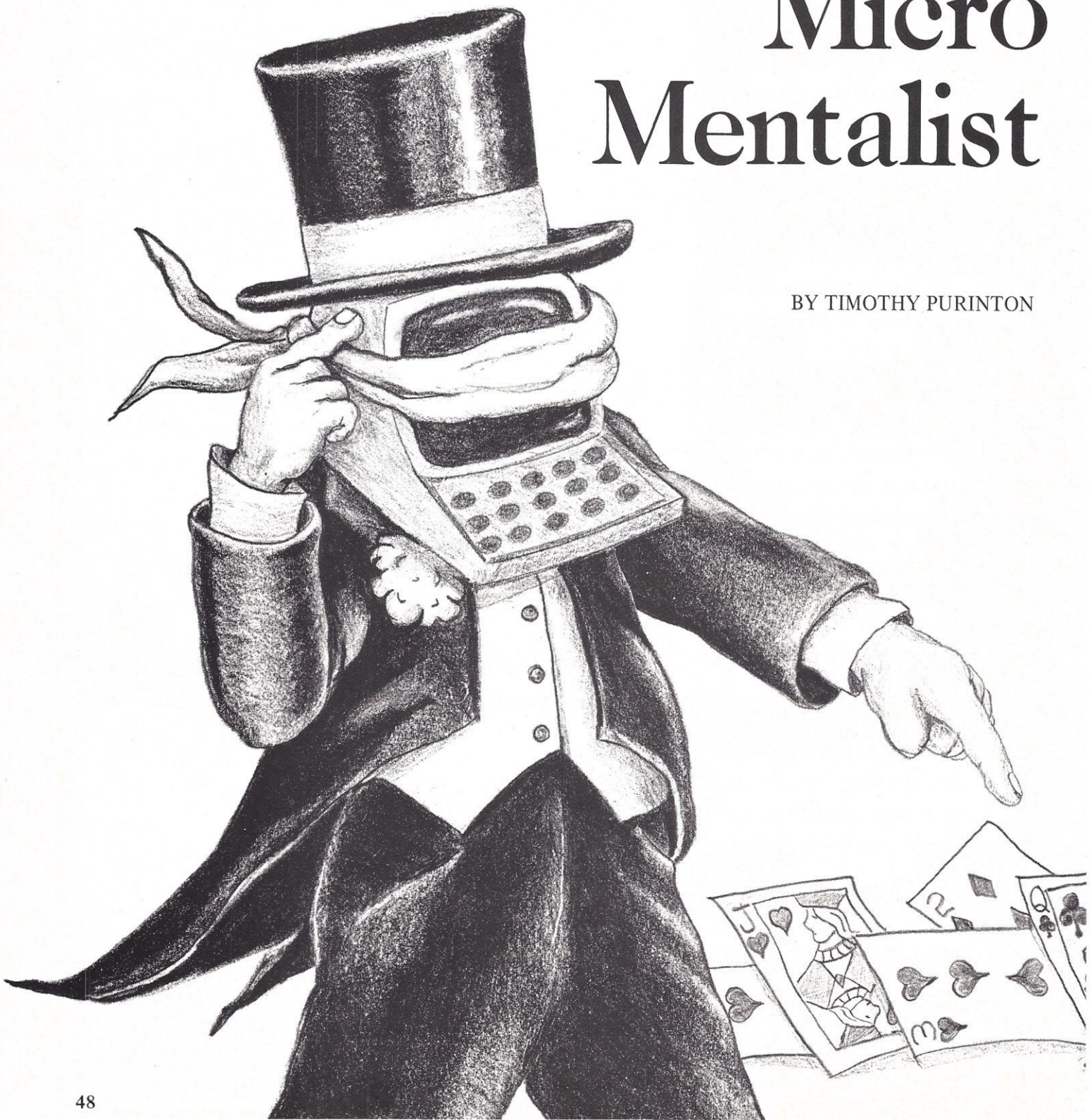
Get details and order now at Radio Shack stores and dealers in the USA, Canada, UK, Australia, Belgium, Holland, France, Japan. Write Radio Shack, Division of Tandy Corporation, Dept. C-046, 1400 One Tandy Center, Fort Worth, Texas 76102. Ask for Catalog TRS-80.

Radio Shack®
The biggest name in little computers

Prices May Vary at Individual Stores and Dealers

The Marvelous Micro Mentalist

BY TIMOTHY PURINTON



Young Stanislaus grabbed his father's sleeve and led him to the keyboard. "Sit down, Pop. The machine will tell you what you're thinking."

"What?"

"C'mon, sit down. It will read your mind."

"OK. So let's see it say 'Enough already.' "

"C'mon." His father sighed and sat. "Type 'RUN.' "

```
RUN
CARD TRICK FOR COMPUTER
PLEASE GIVE ME A NUMBER TO
DESIGNATE THIS TURN.
IN DIGITS, PLEASE ?
```

"Now, type in a number. Any number. It seeds the randomizer so we can repeat this particular run if we want to."

Pop tapped out "61.23" and these lines came up.

```
61.23 ALL RIGHT, NOW.
CHOOSE ANY OF THESE FIRST
TEN CARDS AS YOUR FIRST
'KEY CARD.'  USE ITS VALUE
TO COUNT DOWN TO YOUR
NEXT KEY CARD...AND SO ON
TILL YOU'VE RUN OUT OF CARDS.
YOU'LL COME TO A 'LAST KEY
CARD.'THINK ABOUT THAT ONE,
AND I'LL READ YOUR MIND.
FACE CARDS COUNT AS 5.TO
CONTINUE, TOUCH SPACE BAR.

3/HEARTS          1
4/SPADES          2
2/DIAMONDS        3
3/CLUBS           4
10/SPADES         5
ACE/SPADES        6
6/DIAMONDS        7
9/DIAMONDS        8
KING/SPADES       9
8/DIAMONDS        10
```

"The machine will do a card trick for you. It'll deal you a deck of cards one at a time after those first 10 you see now. You'll get one more card each time you touch the space bar. When you've gone through the deck once, it will be shuffled, and you'll go through the deck one more time.

"Now choose any one of those first 10 cards. Don't tell me or the machine which one you've chosen. That card is your *first key card.* Notice its position in the deck is numbered to make it easy for you to count to your next key card. The cards are valued at their own numbers except for the face cards, which are valued at five. So starting at your first key card, add its value to its position and count down to that total to find your *next key card.* Take the value of that card, add it to its position and find the next key card . . . and so on through the deck twice.

"You're looking for your *last key card.*" Stan continued, "Chances are you'll come to a card, near the end, whose value is too big to let you count it all down before you run out of cards at position 104. When you get there, that's your last key card, and again, don't tell me or the machine what it is. The machine will read your mind and tell you. Got it?"

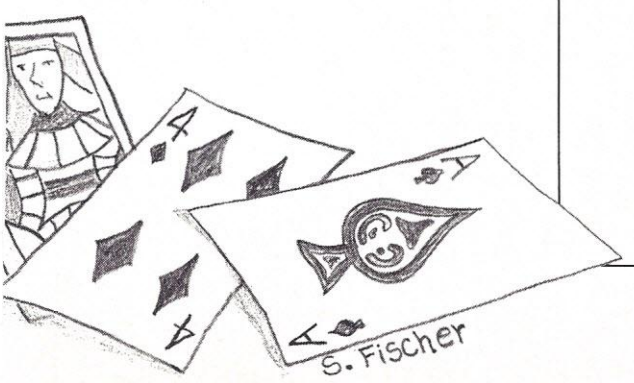
His pop sighed. "I think so. I pick my first key card out of the first ten. I use its value to count down to another key card. Face cards count as fives. I keep going till I find the last key card. And you think the machine will know what it is. So will you, won't you? You'll see me stumbling along on the space bar, trying to count accurately, and you'll figure it out."

"I'll leave the room," Stan said.

(While Stan was out Pop dealt the two decks. See box below.)

While Stan was away . . .

```
10/HEARTS          11
7/CLUBS            12
9/HEARTS           13
2/SPADES           14
QUEEN/DIAMONDS     15
7/DIAMONDS         16
9/CLUBS            17
3/DIAMONDS         18
QUEEN/HEARTS       19
5/DIAMONDS         20
7/HEARTS           21
ACE/DIAMONDS       22
6/CLUBS            23
JACK/SPADES        24
7/SPADES           25
QUEEN/SPADES       26
8/HEARTS           27
8/SPADES           28
4/DIAMONDS         29
5/CLUBS            30
KING/HEARTS        31
ACE/HEARTS         32
10/CLUBS           33
ACE/CLUBS          34
JACK/DIAMONDS      35
5/SPADES           36
JACK/HEARTS        37
9/SPADES           38
JACK/CLUBS         39
4/CLUBS            40
2/HEARTS           41
5/HEARTS           42
2/CLUBS            43
4/HEARTS           44
KING/CLUBS         45
8/CLUBS            46
6/SPADES           47
10/DIAMONDS        48
6/HEARTS           49
QUEEN/CLUBS        50
KING/DIAMONDS      51
3/SPADES           52
3/HEARTS           53
4/SPADES           54
2/DIAMONDS         55
3/CLUBS            56
10/SPADES          57
ACE/SPADES         58
6/DIAMONDS         59
9/DIAMONDS         60
KING/SPADES        61
8/DIAMONDS         62
10/HEARTS          63
7/CLUBS            64
9/HEARTS           65
2/SPADES           66
QUEEN/DIAMONDS     67
7/DIAMONDS         68
9/CLUBS            69
3/DIAMONDS         70
QUEEN/HEARTS       71
5/DIAMONDS         72
7/HEARTS           73
ACE/DIAMONDS       74
6/CLUBS            75
JACK/SPADES        76
7/SPADES           77
QUEEN/SPADES       78
8/HEARTS           79
8/SPADES           80
4/DIAMONDS         81
5/CLUBS            82
KING/HEARTS        83
ACE/HEARTS         84
10/CLUBS           85
ACE/CLUBS          86
JACK/DIAMONDS      87
5/SPADES           88
JACK/HEARTS        89
9/SPADES           90
JACK/CLUBS         91
4/CLUBS            92
2/HEARTS           93
5/HEARTS           94
2/CLUBS            95
4/HEARTS           96
KING/CLUBS         97
8/CLUBS            98
6/SPADES           99
10/DIAMONDS        100
6/HEARTS           101
QUEEN/CLUBS        102
KING/DIAMONDS      103
3/SPADES           104

THAT'S ALL THE CARDS.
I KNOW YOUR LAST KEY CARD.
READY FOR THE REVELATION?
```



Program Notes

INPUT at 3 is, as Stan said, just a seed to permit repetition of the "random" run if it's wanted.

The DIM statement at 4 reserves an array for the oversize subscript. KK gives the computer an arbitrary starting place for its private calculations. The computer is thus given the card at position #1, which here happens to be 3/HEARTS, as its first key card – while Stan's pop is choosing one of his own, not necessarily the same.

Line 15 is one of Stan's mistakes. He meant to delete the line when he moved it to 19. Here it gives the computer a little useless work, but doesn't interfere with the run.

At 18, the randomizing algorithm is seeded.

At 19, 20 and 29 we find the textbook method of "dealing cards" without dealing the same one twice, thus gradually exhausting the deck. You can think it out or look it up. "T=T+1" counts all cards dealt.

Stan says there may be a more efficient way to identify the randomly-generated cards than by the long sieve at 30-82, but if there is he hasn't found it. Maybe you can do better. The "A=" statement within these lines is the computer's private evaluation system. (See "How the trick works".)

Line 5300 is a counting-variable, which works at 5340 to let all of the first ten cards be dealt without pause.

At 5320, Stan activates the computer's private calculation leading to the "last key card". The sub-routine at 8000 says, "Starting at 1 – the value we've given to KK, and the first card dealt – build up the value of KK by the value of the card at that position." On the first pass in this case, KK becomes 4.

At 8010, we say, "Since we're using not more than 104 cards, discount the value of any final key card too big to be counted out, and remember its name, Z$, for the revelation." At 8020, we go back to print the names and positions of cards called up at 5330.

The counting-variable T at 5335 stops the deal after we've gone through the deck of 52 twice.

The PEEK-POKE pair at 5360-5361 stops the run to let Stan's pop continue to deal at his own pace by touching the space bar – a handy technique for many applications.

At 5362 and in the succession of sub-routines, the program sends the run through the deck a second time after the first 52 cards have been dealt and sends it to its finale after the second 52. The ZZ and XX statements at 5800 and afterwards are simply for time-lag. Stan likes to make the computer act like a person.

Program Listing

```
1 PRINT"CARD TRICK FOR COMPUTER"
2 PRINT"PLEASE GIVE ME A NUMBER TO DESIGNATE THIS TURN."
3 INPUT "   IN DIGITS, PLEASE "; NX:CD=NX
4 DIM L(52):KK=1
5 PRINT"ALL RIGHT, NOW.  CHOOSE ANY OF THESE FIRST TEN CARDS"
6 PRINT"AS YOUR FIRST 'KEY CARD.'  USE ITS VALUE ";
7 PRINT"TO COUNT DOWN TO YOUR NEXT KEY CARD...";
8 PRINT"AND SO ON TILL YOU'VE RUN OUT";
9 PRINT" OF CARDS.  YOU'LL COME TO A 'LAST KEY CARD.'";
10 PRINT"THINK ABOUT THAT ONE, AND I'LL READ  YOUR MIND. ";
11 PRINT"FACE CARDS COUNT AS 5.";
12 PRINT"TO CONTINUE, TOUCH SPACE BAR."
15 FOR I=1 TO 52: L(I)=I: NEXT I
18 K=RND(-NX)
19 FOR I=1 TO 52: L(I)=I: NEXT I
20  FOR J=52 TO 1 STEP -1:R=INT(J*RND(1))+1: T=T+1
29 X=L(R): L(R)=L(J)
30 IF X=1 THEN X$="ACE/SPADES" :A=1: GOTO 5300
31 IF X=2 THEN X$="2/SPADES": A=2: GOTO 5300
32 IF X=3 THEN X$="3/SPADES": A=3: GOTO 5300
33 IF X=4 THEN X$="4/SPADES":A=4: GOTO 5300
34 IF X=5 THEN X$="5/SPADES":A=5: GOTO 5300
35 IF X=6 THEN X$="6/SPADES":A=6: GOTO 5300
36 IF X=7 THEN X$="7/SPADES":A=7: GOTO 5300
38 IF X=8 THEN X$="8/SPADES":A=8: GOTO 5300
39 IF X=9 THEN X$="9/SPADES":A=9: GOTO 5300
40 IF X=10 THEN X$="10/SPADES":A=10: GOTO 5300
```

In a minute or so, his pop yelled, "I'm ready." Stan came back. "The computer will never get it," said Pop. "My last key card could be any of the last several cards, couldn't it?"

"Could it?" said Stan. "Tell the computer you're ready for the answer."

His pop touched the keyboard and looked at the CRT.

```
YES
IT'S QUEEN/CLUBS
SHALL WE TRY IT ANOTHER TIME?
```

"Ugh," he said.

```
NO THANKS
HOPE YOU WERE PROPERLY BAFFLED.
BYE-BYE.  (THAT WAS   61.23 )
```

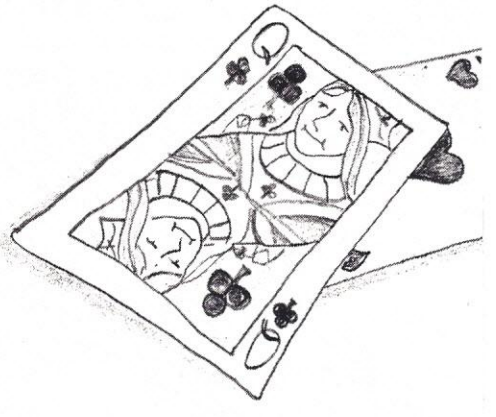
He stood up and edged away. "Can't fool around here all day. I've got things to do."

Stan grinned. "Glad you liked it."

How the trick works

Oh. You understand how it works, don't you?

In the run shown, the "computer's private calculation" started at position #1 with 3/HEARTS, an A-value of 3. KK, which starts out at 1, adds A to become 4. When T, the card-position counter, becomes equal to KK at 4, this A-value is added to KK, so in this case

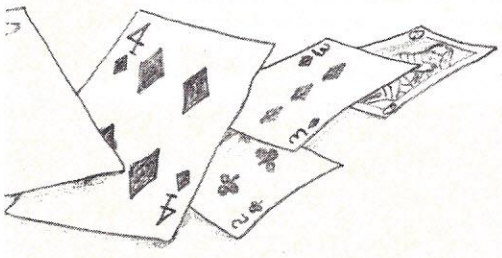


the 3/CLUBS adds 3 and KK is now 7. At position 7 we find 6/DIAMONDS, and add 6 to KK. This process continues as the computer runs through the deck twice. The checkpoints in this deal are 13, 22, 23, 29, 33, 43, 45, 50, 55, 57, 67, 72, 77, 84, 85, 95, 97, 102 . . . and out, since the "QUEEN/CLUBS" at position 102 is valued at 5, and 102 + 5 adds up to more than 104. So the run gets a ZS as "QUEEN/CLUBS", ready to be revealed when Stan's pop reluctantly types his final "YES".

Meanwhile, Stan's pop has been making his own calculation. Suppose he chose as his first key card the ten of spades, at position 5. His checkpoints would go: 5, 15, 20, 25, 32, 33 . . . ah ha! An intersection! Position 33 is one of the computer's private checkpoints, so from here on the computer and Pop will be in step all the way to the QUEEN/CLUBS. Suppose Pop starts at 9/DIAMONDS; the intersection occurs at 43. Suppose he starts instead . . . but check it out.

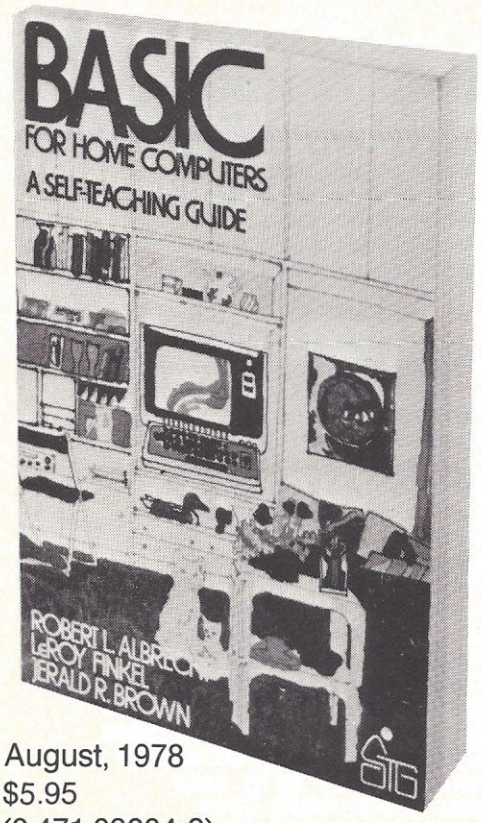
It's possible to miss intersecting checkpoints with a single pass through a 52-card deck, though the probability of success even that way is reckoned at 5/6. With 104 cards, can you miss? You might try to calculate your chances.

Try it on your pop. He may not speak to you for days. □



Program Listing Continued

```
41 IF X=11 THEN X$="JACK/SPADES":A=5: GOTO 5300
42 IF X=12 THEN X$="QUEEN/SPADES":A=5: GOTO 5300
43 IF X=13 THEN X$="KING/SPADES":A=5: GOTO 5300
44 IF X=14 THEN X$="ACE/HEARTS":A=1: GOTO 5300
45 IF X=15 THEN X$="2/HEARTS":A=2: GOTO 5300
46 IF X=16 THEN X$="3/HEARTS":A=3: GOTO 5300
47 IF X=17 THEN X$="4/HEARTS":A=4: GOTO 5300
48 IF X=18 THEN X$="5/HEARTS":A=5: GOTO 5300
49 IF X=19 THEN X$="6/HEARTS":A=6: GOTO 5300
50 IF X=20 THEN X$="7/HEARTS":A=7: GOTO 5300
51 IF X=21 THEN X$="8/HEARTS":A=8: GOTO 5300
52 IF X=22 THEN X$="9/HEARTS":A=9: GOTO 5300
53 IF X=23 THEN X$="10/HEARTS":A=10: GOTO 5300
54 IF X=24 THEN X$="JACK/HEARTS":A=5: GOTO 5300
55 IF X=25 THEN X$="QUEEN/HEARTS":A=5: GOTO 5300
56 IF X=26 THEN X$="KING/HEARTS":A=5: GOTO 5300
57 IF X=27 THEN X$="ACE/DIAMONDS":A=1: GOTO 5300
58 IF X=28 THEN X$="2/DIAMONDS":A=2: GOTO 5300
59 IF X=29 THEN X$="3/DIAMONDS":A=3: GOTO 5300
60 IF X=30 THEN X$="4/DIAMONDS":A=4: GOTO 5300
61 IF X=31 THEN X$="5/DIAMONDS":A=5: GOTO 5300
62 IF X=32 THEN X$="6/DIAMONDS":A=6: GOTO 5300
63 IF X=33 THEN X$="7/DIAMONDS":A=7: GOTO 5300
64 IF X=34 THEN X$="8/DIAMONDS":A=8: GOTO 5300
65 IF X=35 THEN X$="9/DIAMONDS":A=9: GOTO 5300
66 IF X=36 THEN X$="10/DIAMONDS":A=10: GOTO 5300
67 IF X=37 THEN X$="JACK/DIAMONDS": A=5: GOTO 5300
68 IF X=38 THEN X$="QUEEN/DIAMONDS":A=5: GOTO 5300
69 IF X=39 THEN X$="KING/DIAMONDS":A=5: GOTO 5300
70 IF X=40 THEN X$="ACE/CLUBS":A=1: GOTO 5300
71 IF X=41 THEN X$="2/CLUBS":A=2: GOTO 5300
72 IF X=42 THEN X$="3/CLUBS":A=3: GOTO 5300
73 IF X=43 THEN X$="4/CLUBS":A=4: GOTO 5300
74 IF X=44 THEN X$="5/CLUBS":A=5: GOTO 5300
75 IF X=45 THEN X$="6/CLUBS":A=6: GOTO 5300
76 IF X=46 THEN X$="7/CLUBS":A=7: GOTO 5300
77 IF X=47 THEN X$="8/CLUBS":A=8: GOTO 5300
78 IF X=48 THEN X$="9/CLUBS":A=9: GOTO 5300
79 IF X=49 THEN X$="10/CLUBS":A=10: GOTO 5300
80 IF X=50 THEN X$="JACK/CLUBS":A=5: GOTO 5300
81 IF X=51 THEN X$="QUEEN/CLUBS":A=5: GOTO 5300
82 IF X=52 THEN X$="KING/CLUBS":A=5: GOTO 5300
5300 C=C+1
5320 GOSUB 8000
5330 PRINT TAB(0);X$; TAB(15);T
5335 IF T=104 THEN PRINT: PRINT "THAT'S ALL THE CARDS."
5340 IF C<10 THEN 5370
5360 IF PEEK(-4611)<>32THEN 5360
5361 POKE -4611,0
5362 IF J=1 THEN GOSUB 6000
5370 NEXT J
5400 GOSUB 7000
5401 INPUT "ANOTHER TIME";C$
5420 IF LEFT$(C$,1)="Y" THEN 5800
5450 PRINT"HOPE YOU WERE PROPERLY BAFFLED."
5460 GOTO 5999
5800 PRINT "I'LL SHUFFLE.": FOR ZZ=1TO 2000
5801 NEXT ZZ: PRINT"I'M SHUFFLING"
5802 FOR XX=1 TO 5000: NEXT XX:PRINT"I'VE SHUFFLED."
5803 PRINT"THAT DESIGNATING NUMBER WAS "; NX
5804 PRINT"PLEASE 'RUN' AGAIN"
5805 PRINT"WITH A DIFFERENT NUMBER."
5806 STOP
5999 PRINT "BYE-BYE.  (THAT WAS  "; CD;") ": END
6000 D=D+1: IF D>1 THEN 5400
6005 GOTO 15
6010 RETURN
7000 PRINT: PRINT"I KNOW YOUR LAST KEY CARD."
7010 INPUT "READY FOR THE REVELATION"; D$
7020 IF LEFT$(D$,1)="Y" THEN PRINT "IT'S "; Z$
7030 PRINT: PRINT"SHALL WE TRY IT ";
7040 RETURN
8000 IF T=KK THEN KK=KK+A
8010 IF KK>104 THEN KK=KK-A:Z$=X$
8020 RETURN
```



Here it is!

Your self-teaching guide to BASIC FOR HOME COMPUTERS

by Bob Albrecht

and **LeRoy Finkel,** and **Jerald R. Brown,**
all of Dymax Corporation.

August, 1978
$5.95
(0 471 03204-2)

Whatever your interest—hobbies or applications for home, school, or office—this book will give you the BASIC skills to use with the new personal-size microcomputers. With this book, you'll be able to read, write, and understand the programming language BASIC. You'll not only be able to write your own programs, but you'll also learn how to modify others' programs for your own particular purposes. The authors have been active in the personal computing movement from its beginning, as founders of the People's Computing Company and later of the Community Computer Center and also as teachers and contributors to a wide range of personal computing courses and publications. They've designed this book with your needs in mind—and with their own unique brand of humor—so that no special background is required. You don't even need access to a computer. You'll find that BASIC FOR HOME COMPUTERS gives you all you need to know about making microcomputers work for you!

3 more first-rate Self-Teaching Guides—

BACKGROUND MATH FOR A COMPUTER WORLD
by **Ruth Ashley** of *Duotech* in consultation with **Nancy B. Stern**
Provide yourself with the math skills you need to survive and be effective in the computer world.
286 pp. $4.95 (0 471 03506-8)

ANS COBOL
by **Ruth Ashley** of *Duotech* in consultation with **Nancy B. Stern**
Teach yourself how to write a COBOL program and compile it effectively, paying proper attention to spelling, punctuation, and format.
242 pp. $4.95 (0 471 03507-6)

INTRODUCTION TO DATA PROCESSING
by **Martin L. Harris,** in consultation with **Nancy B. Stern**
It'll show you how computers work, what data processing is, how it's organized, and what types of equipment are used.
326 pp. $4.95 (0 471 35300-0)

Look for these and other STGs at your favorite bookstore. For more information, and a complete listing of current and forthcoming STGs, write to George Carroll, Dept. A3226-64

JOHN WILEY & SONS, Inc.
605 Third Avenue
New York, N.Y. 10016
In Canada: 22 Worcester Road, Rexdale, Ontario

Prices subject to change without notice. A3226-64

CIRCLE 16

The TRS-80 Comes to Town

BY HARRY SHERSHOW

Why all the interest in the TRS-80? On June 8, 1978, more than 1,000 inquiring prospects came to Boston's Sheraton Hotel to see Radio Shack's presentation of the TRS-80 microcomputer. The show was the 16th in a series of 26 being held around the country.

"Why all the interest in the TRS-80?" echoed Jerry Colella, regional manager of Tandy Corporation, parent company of Radio Shack. "Because it requires less money to get started in this activity than it does to buy a decent stereo component system. Right now, as a matter of fact, we're selling in Radio Shack stores stereo components that cost almost twice the $600 required to buy the basic TRS-80 unit. And let's face it. The microcomputer era arrived a year or two ago. It's only a question of time until everyone who owns a television will also have a computer, either as a system hooked into the TV or as a stand-alone system like our TRS-80. Tomorrow is here today!"

The demonstration at the hotel was similar to most such shows. Coffee on a side table for thirsty sightseers, chatty salesmen at everyone's elbow, more than 150 TRS-80 micros on tables with entreaties from the ushers to "Play with 'em! Bang 'em! Kick 'em! Do anything you want – you can't hurt them!" Of course they discourage sledge hammers and vandals pouring water in the keyboard. What they're trying to illustrate is that the TRS-80 is a rugged instrument which unlike your $1500 stereo amplifier, can't be hurt by rough handling.

So throngs of viewers whacked the keyboards and pounded the buttons and apparently had an enjoyable afternoon not only listening to sales pitches of the attendants, but also sitting through a session in which they quickly learned how to write a complete program in BASIC.

Personal Computing's correspondent offered himself at the show as a sales prospect to young, affable Al Jones, manager of the Chestnut Hill, Mass., Radio Shack.

Personal Computing: Al, Let's make believe I'm a farmer, just in from the farm. Know nothing about computers. I'm here to learn everything about these.

Al Jones: Okay. We'll start at the beginning. You are sitting in front of a TRS-80 microcomputer. It is a complete unit as you see it: that video display, which looks like a television screen, and the keyboard, which looks like a typewriter. The computer itself is sealed inside the keyboard. With the attached cassette player it is complete. Nothing else needed to make it run at its primary level which is Level 1. Talk to the computer through the keyboard and it talks back to you through the video screen.

PC: Talk?

Al: It doesn't actually talk, although eventually voice synethizers will be added to upgraded models. Right now, all it does is print statements. This computer is a good deal like a calculator. After it's done something, it will remember what it's done as long as

you keep the machine on. If you turn it off, everything you have entered through your keyboard is lost. Goes out like a light when you throw the switch.

PC; You could waste a lot of time if you accidently pushed the wrong switch after you've worked on it all day.

AI: Well, we've provided for that. We have here at the side, as you can see, a portable tape recorder which is hooked into the machine. A cassette in the tape recorder retains your programs; you can't erase them no matter what switch you throw by mistake. With the use of the cassette recorder, we provide you with a complete package so that, as a beginner, you don't have to learn anything about the computer. The package includes a user's manual which is as simple to read as the comic strips. The 250 pages in it are written on the premise that you know absolutely nothing about the computer. After you finish reading the manual you will be able to make the computer perform every command available at this level of computing.

PC: What kind of commands are those?

AI: There are actually only six commands to make your machine start working.

PC: Only six? Seems like a pretty low minimum.

AI: Those are commands like LIST, STOP, RUN. Just the simple commands to activate the machine. Like whipping a horse to get it moving. Actually, of course there's more to it than the commands. There are 16 program statements, two print modifiers, and four graphics statements.

PC: The six commands seem to have mushroomed fast.

AI: I've just told you everything – absolutely *everything* you will have to know to get the computer working for you. I told you that the six commands were like whipping a horse when you're sitting in the buggy. Well, what good is it to whip the horse and get it going unless you tell it where to go? You have to know how to steer that horse so it won't just stand there and run in one place. From the minute you get the crate from Radio Shack unpacked, the user's manual will show you how to set up the machine and how to get it running in a matter of minutes.

PC: It sounds as though you've simplified it quite well.

AI: Very simple. We even teach you a language. BASIC. You only learn a little at a time. Just the things you have to learn: the six commands and two or three of the program statements. Of course, the more you learn, the better your computer performs. Simple as that.

PC: What about these cassette programs?

AI: The cassettes are no different than a tape recorder. Of course they can't be programmed or erased. They're permanent. Which is a good thing. Later on, you will learn how to make changes on 'em if you have to. Once you learn how to make your own programs you won't bother with cassettes so much. Eventually, you're going to get more interested in what the computer is going to do for you than you are in the cassettes. So you do your own programs. Say you have a list of customers who buy milk, eggs, beef and produce from you – if you're a farmer – and you want to keep them separate. You want to know how much money they owe you and how much they buy. Even want to be able to send them bills at the right time or to remind them they haven't paid their bills yet. This computer, once you understand it will do all that for you. All you are going to have to do is learn those few words we mentioned and that's all it will take to run your computer. Let me tell you a story of a case in my store. I have a customer, a doctor who deals with many patients. He has about 6000 patients and decided to put them on a computer. I let him read the user's manual and he became an instant programmer and bought the

TRS-80 microcomputer specifications

HARDWARE

Microprocessor: Advanced Z-80 8-bit processor.
Keyboard: Integrated ASCII, 53-key professional-type.
Video Display: Memory mapped, all graphics and alphanumerics controlled by BASIC commands. Automatic scrolling.
Text: 16 lines of 64 characters, also software selectable to 32 characters per line.
Graphics: 128 horizontal by 48 vertical. Graphics and text can be interspersed in any manner by software.
Memory: Includes 4K Read-Only-Memory (ROM), 4K dynamic Read/Write Memory (RAM). Internally expandable in the computer case to 12K ROM and 16K RAM. Total memory capability to 62K.
Input/Output: Computer-controlled cassette interface. Expansion port for additional memory and peripherals. Keyboard built-in.
Electrical: U.L. listed for 120 volts AC, 60 Hz.
Dimensions: 16½ x 8 x 3½".

SOFTWARE

Radio Shack Level-I BASIC in ROM.
Level 1 Features: standard BASIC statements; floating point arithmetic; numeric, array and string variable; video graphics commands; cassette save and load commands.
Commands: NEW, LIST, RUN, CONTinue, REMark, LET, FOR-NEXT-STEP, GOSUB-RETURN, STOP, END, GOTO, IF-THEN, INPUT, ON . . . GOSUB, PRINT, CSAVE, CLOAD, DATA, READ, RESTORE.
Functions: MEM, TAB, INT, ABS, RND, +, –, *(multiply), /(divide), <, >, =.
Special Commands (Including graphics): CLS (clear screen), SET (x,y), RESET (x,y), POINT (x,y), formatted PRINT. Array and string capability. Data storage and retrieval.

OPTIONS

Video Display: 12" diagonal screen. 16½ x 13½ x 12". U.L. listed.
CTR-41 Data Cassette: Battery or AC operation. U.L. listed

TRS-80. Now he types his own information concerning the patients' cases so he can keep track of them. He has the program running pretty well now. He's so pleased with his TRS-80 that he's going to grow with it. He feels he's used up the computer in its present state. He started by learning those simple commands I told you about. Now he's become a pretty good novice and he's ready to expand. That's the idea of TRS-80. Start small and grow with it as you go along.

PC: Can he also do billing with it?

AI: He will use it for billing or for any other thing he wants. It's part of the growing process. The machine is expandable. It's the one big thing you can say about the TRS-80. Buying this small machine does not necessarily mean that you cannot enlarge it to our full-size business system. We are basing the full-size system around these same components. The only thing that's expanded is the size of the memory and the language. Just those two things. Everything is located on only one board, right now. As it sits here, the way you see it, it has only 4K of read-only memory. So, increasing memory capacity, which is what we're doing when we upgrade our machine, means simply going from 4K to 16K or more. The machine stays the same size. Actually if you looked at two of our machines you couldn't tell which was 4K and which 16K. A 16K machine runs twice as fast. All you do is bring this original keyboard back to the store. We then send it to our service center where they stick in three more memory chips and presto, – you've got a teen-ager in the family! A 16K memory computer.

PC: How long does the conversion take?

AI: A couple of days. But counting the time it takes to get to our local service center and back it could be a week or two.

PC: Except you have to go back to your store all the time. Do you suppose customers might consider that a catch?

AI: There's no catch. We do this so we can honor our warranty. If you try to do it yourself and burn out a couple of chips or spill hot solder over the circuit board and ruin it, who's going to fix it? This way *we* stand behind the machine all the time, within the warranty. The machines are sealed as you can see, so no one but our own experts can touch them. We don't want you tampering with the machines. A good product is our best advertisement. The chips are easy to destroy. Now when you get around to expanding we have all sorts of peripherals like screen printers, line printers and so forth. And more in the planning stage. If you want bills, invoices, paychecks and things like that, you'll *have* to get the line printer.

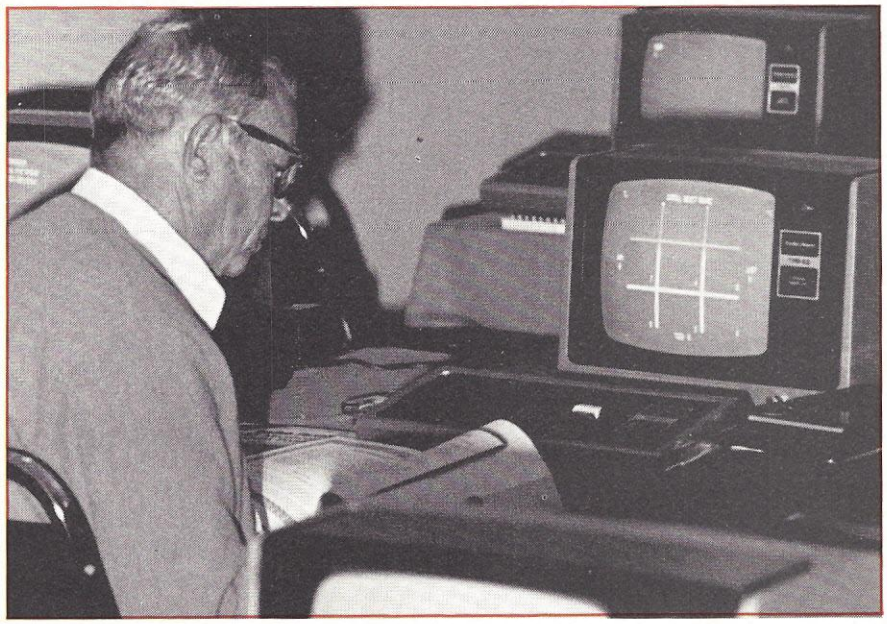
PC: Can you use the line printer on *this* machine?

AI: No. You've got to upgrade it to Level II. That's because the printer requires certain other commands which are not in the Level I vocabulary. You have to have Level II. You'll also need the expansion interface.

PC: Thought you could hook right into ports on the keyboard?

AI: You can with the screen printer or cassette recorder. The interface is simply a box which plugs in by ribbon cable from the main computer to the peripheral. The interface has facilities

"You get a portable tape recorder that hooks into the machine. With this cassette recorder you have a complete package and you don't have to learn anything about the computer."



"If your computer fails for some reason and has to be sent back for servicing, you simply bring it into your local Radio Shack - and there are more than 5000 across the country."

for four ports inside. So you can have a combination of what to use. Say, for instance, you didn't want a printer; you just want a lot of bulk memory to print out on the screen. The screen will handle up to four floppy disks. I use floppy disks in my own store right now, as a small business system. It does inventory, payroll, general ledger and profit-and-loss statements as well as names and addresses for our regular mailings.

PC: Still all with that BASIC language? It seems to me BASIC is limited.

Al: Well, you bring up a good point. Some machines are stuck on their own language and that's all they can use. Radio Shack has recognized the fact that there are many languages. obviously from a novice's point of view, BASIC is the easiest to learn. But as you grow with the computer you may someday decide that there are other languages you may want to use – for one reason or another. But we can handle that with this machine. For exple, we have an assembly language tape. It literally bypasses the address code – all those commands and statements you see in the manual – for BASIC. This assembly language tape puts you right into pre-memory so you can program the machine directly. You're bypassing BASIC altogether. Now, suppose you're familiar with the languages of FORTRAN or COBOL. Well, we have those languages on our mini-disk system and again you can bypass the BASIC and work in the language you're most confidant in.

PC: That means you will have to know the FORTRAN language?

Al: Oh yes. You'll have to know FORTRAN to use it just as you would have to know BASIC. But as the machine sits here right now, it's strictly a Level I computer. It deals in BASIC language only. But you don't have to sit down and learn the language. All you have to do is learn those few commands and statements you see in the user's manual. It would take you longer to set up and operate a stereo-component system where you don't have to use any language at all. Stereo components are much more complex physically than this simple computer.

PC: The manual looks easy enough to read.

Al: This manual, I would say, is the easiest, most instructive one on the market today. It takes you from *zero* – no knowledge at all on your part – right through all the commands. It gives you definitions of those commands and statements so that you know what they mean and what they do. Then it gives you easy examples that teach you quickly what they *do*. As you move along through the manual you get examples of how to combine groups of commands to run simple game programs and business programs.

PC: Like standing on a diving board. You get the impetus to dive and go off on your own.

Al: That's right. We find that beginners can't wait to get to that part. Obviously, you are limited as to how far you can go in your own programming without upgrading. Just as in the case of the doctor and his patients' records we were talking about a little earlier. His program is not in this manual. But the basic ingredients of running the program and learning the print statements of the program are in there. Just a question of taking the right steps out of the manual, combining them and making up his own program. And that is where you leave Level I and get into Level II. When you're ready for that, the Level II manual does the same for you – in it's easy, chatty style – that Level I does for you now. Once you're familiar with Level I, then Level II is a snap. You should have no trouble working with it.

PC: From what you say, it sounds rather easy.

Al: Let me put it this way. Personal computing, from my own standpoint, is a learning process. It's a hobby like anything else and you advance as far as you want. Some people will obviously stay with Level I and not want to go any further. They will find it serves their needs and they'll have no use for Level II. Other people, though, will be building full-blown systems for their homes or for their business or for special learning applications in whatever language is convenient for them as quickly as they can.

PC: I suppose after someone has used TRS-80 for some time he will be in a position to consider himself a professional programmer?

Al: That's another thing. Owners and constant users of this computer – or any other computer for that matter – are going to become skilled. And the feeling is that there's going to be a huge demand for skilled programmers in industry.

PC: There is a demand right now.

Al: Well, that's nothing to the demand that's coming, the way I see it. Pandora's box is open and everything is leaking out. Growing like mad.

PC: All you have to do is look in the help-wanted section. Every other ad is for a computer programmer or a computer designer. It's just amazing. I don't remember any time in the history of the country where there was such an advertised demand for one particular scientific specialty. Sure, there have been demands, in the past, for aerospace engineers, sonic engineers, and so forth. But never for one particular class of professionals – computer employees. But the type of programming they're looking for, you can't learn by simply working on the TRS-80. They want people familiar with ALGOL, COBOL, FORTRAN and others.

Al: True. But remember we've been talking about micro-programming. You can go easily into advanced stages. It depends on yourself. Look what I'm doing with my TRS-80 in my own store. And I've never studied programming or computer science. I wrote a profit-and-loss statement for my store – and there's nothing in the manual that tells you how to do that. You have to go out and learn to do it yourself. Just shows that it's not hard to advance. In other words you can't swim unless you get in the water and the longer you stay in the water the better swimmer you become. For that P-and-L statement, of course, you have to know what a profit-and-loss statement is and what information you want to get into it. If you know the commands of the TRS-80 and you know where things are going to go, then you just sit back and let the computer use its mathematical ability to figure things out for you. With this machine I can figure out what daily sales are, what it costs me to stay in business, my overhead and my profit for the week. And that is, in fact, a business program that anyone can do for himself or go into a little business doing it for other people.

PC: About price. Everyone, of course, is interested in price. That is Radio Shack's biggest selling point. Right now, you're offering the keyboard, the VDT and the cassette as one unit. You have to buy all these pieces to get started?

Al: Not really. We've broken it down to simpler components. Of course we'd like people to buy the cassette because we know they'd get a heck of a lot more enjoyment out of their computer with it. However we're going on the assumption that the person buying his first computer is a strict novice and has absolutely no background at all in computing. So we offer him a simple, complete, elementary set-up: the keyboard with computer built-in, the video-display and the cassette recorder all for $599. If you bought the cassette separately it would cost you around fifty dollars. But you get it free when you order the display unit with the keyboard. You can buy the CPU separately, too, for $400 if you already have a video display.

PC: What about that upgrading you mentioned?

Al: You can get a Level II BASIC ROM for $99 and a 16K RAM for $290.

PC: What about peripheral prices?

Al: A line printer is $1290; a screen printer, $599; a 16K mini-disk, $499; and an expansion interface so you can plug the printer and mini-disks into the system is $299. But we're not talking about all these things at once. It's something you can think about as you grow with your basic unit. Something for future upgrading. It's a lot like stereo components. Start with amplifier. Later buy a turntable; then a multiplex receiver; then a cassette recorder. A full-blown stereo system can add up to a couple of thousand dollars, but you start only with the basic units and work your way up.

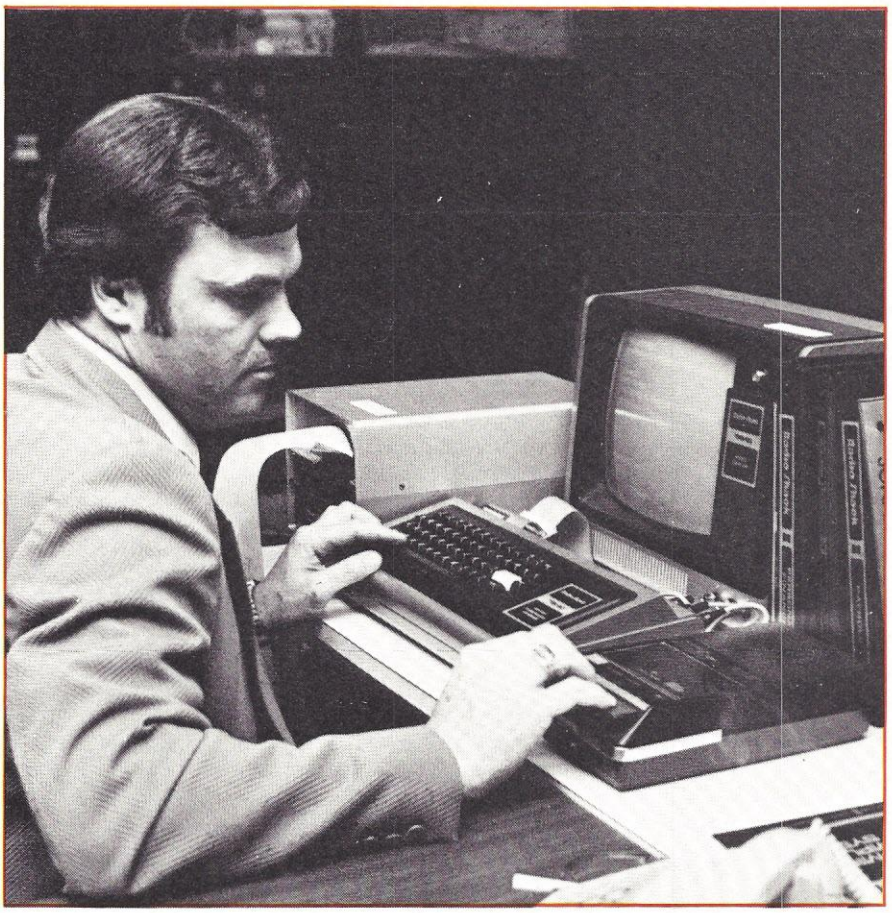
PC: What about software?

Al: As of now we have available Math 1, Algebra 1, Personal Finance, Home Recipes, Statistical Analysis – about 12 in all. But we're expanding that line. There's a chess game on cassette that's coming out. It allows you to play chess against your computer and try to beat it if you can. Usually you can't at the higher levels of chess – unless you're a pretty good player.

PC: What about delivery?

Al: On Level I you can get a unit right now. Pick one off the table and walk out with it. Paying for it first, though. Or you can come to my store where I have them in stock. You can even have Level I in 16K instead of 4K for another $300. There's a waiting period for delivery of two weeks on Level II. But we're not as bad as some of the computer companies that show you a computer and don't deliver for a year or so. If you want to order from a Radio Shack store and if you want a model from the factory you can have

Radio Shack computer owners who would like to exploit the capabilities of their video display should know about an easily-performed modification conceived by Dan Likins of Santa Ana, CA. Without loss of TRS-80 graphics, the procedure allows display of all upper and lower case letters, numerals, all control and special characters. You need only install two jumpers about 2" long, cut one trace and add an additional 2102 static memory. Details are in the Computer Information Exchange Newsletter, which is free. Write CIE, Box 158, San Luis Rey CA 92068. Enclose label and 15¢ postage.



one with a reasonable wait. Our factory is putting in a whole lot of new production lines and they are increasing their own efforts to keep up with the increased demand. Peripheral delivery time, for example, is running approximately 10 weeks. But as production and sales keep going up, as they are doing, those times will certainly be greatly reduced.

PC: How long a wait if something goes wrong with the computer and you have to send it back for repairs?

AI: That's a point we haven't covered, and it's pretty important. We are different from other computer companies because our service department is not back at the main factory. If your computer fails for some reason and has to be sent back for servicing, you simply bring it into your local Radio Shack – and there are more than 5000 across the country. From the store it is sent to a regional service center, not all the way across the country. Every area has its own service center so you can see we have organized ourselves to serve the consumer better than anyone else. What other computer company has service centers all over the country and a place where you can walk in with your sick computer in your arms practically in your own neighborhood, or wherever the nearest Radio Shack is located?

Announcer on Stage: Welcome to Radio Shack's display of the TRS-80. It will be my pleasure to talk to you about the computer, answer as many of your questions as I can. At the same time, we are all going to write a program on the computer to give you an idea of how relatively simple it is. The program will show you the sequence in which information typically is entered and it will show you some math calculation formulae and it will show you how important it is to enter the information correctly.

AI: He's going to talk for about half an hour. By the time he's finished, everyone sitting here at the computer will be able to type in a program.

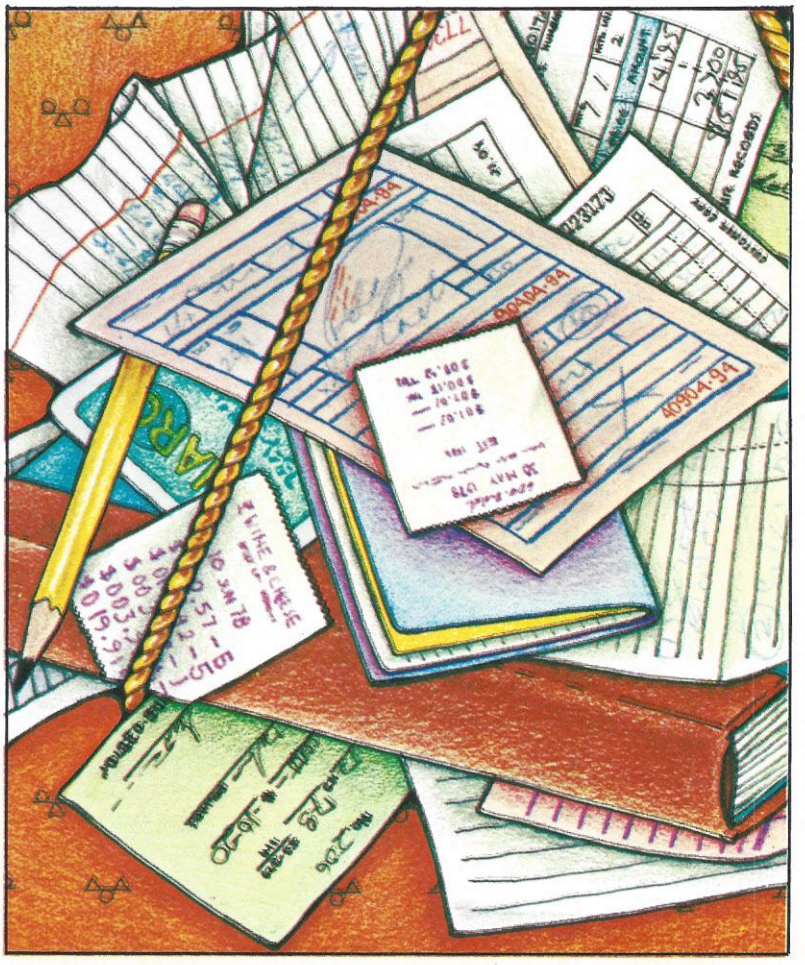
PC: I think I'll run along now.

AI: Can I sell you a computer? Only $599 while they last.

PC: Let me think about it. I want to talk to someone who owns one. □

The Computer Checks into the Balancing Act

BY O. E. DIAL



The most significant and revolutionary development in personal computing (other than cost) is the turnkey operation of microcomputers. Computers can be operated by individuals who have absolutely no training in computer science – just as you don't need to be an auto mechanic to be a competent driver.

But before the full potential of personal computing can be realized, the software must be user-oriented. Programs must speak to the user in ordinary English, prompting where choices must be made and providing a range of choices from which to select commands to the computer.

Further, the software must anticipate the input errors the user is most likely to make and, where possible, detect such errors before they've been acted on, alert the operator to the error and then allow the user to correct the input.

This program (which will be continued in subsequent issues) was designed with the user-orientation concept in mind. The system is intended for the layman, not only with respect to computer operation, but with respect to accounting as well.

CRAS

CRAS – Check Register (checkbook) based Accounting System – is fully integrated so a single transaction entry is all that's required to produce a large array of specialized statements, including Check Register, Check Register Notes, Accounts Distributions, Statements of Selected Accounts, Checkbook Reconciliation and the printed check itself.

The system was originally designed for individuals who conduct both business and personal affairs from a single checkbook, and who have one eye on the level and arrangement of detail needed for income tax preparation at the end of the fiscal year. The system yields a printed statement for the month(s) or part of a month desired. This statement reflects the details of each transaction, the resulting balance and a division between business and personal entries. However, the system may be used without change exclusively for business purposes, or exclusively for personal entries.

CRAS provides for 36 accounts (#10-#45) with user-selected titles for each. In its present form, 16 accounts are reserved for business and 21 accounts for personal use. Of the business accounts (#10 through #25), one account (#25) is reserved for business income. Of the personal accounts (#26 through #45), three are reserved for income (Accounts #43 through #45). All remaining accounts are pegged as expense accounts. The number of accounts reserved as income accounts may be altered by changes indicated later in this article.

This program was designed for naive users, and leads you through each step of the operation. People with no training in computer hardware, computer software or accounting, can use CRAS. All technical activity is behind the scenes. You simply respond to the video terminal, answering questions about checkbook transactions, and indicating what kind of statement to print. The computer then produces whatever ouputs you wanted.

When I first began using CRAS, I kept an ordinary checkbook in parallel with the system, unloading the checkbook to the system once each month. Since that time, I've found it more convenient to abandon the checkbook, merely keeping notes from day to day, and unloading these notes to CRAS when convenient. Of equal importance, the check register

has never failed to reconcile with the bank statement since I began using CRAS about a year and a half ago.

Five lay users tested the system for over one year. The critical comments of these users were noted and the system improved in response. Further, I noted common input errors and, where possible, revised the programming to recognize such errors, to alert the user to mistakes and to give an opportunity to re-input the data before it was acted on.

A CPA (the one who does my own income taxes) also reviewed and critiqued CRAS. His critical comments resulted in changes which improved the system. But generally, when, for the past two years, I appeared in mid-January (!) with a package of printed statements – one for each tax-relevant account – he simply smiled. His task was easy. All the data was there, well organized for easy access.

Originally, the system was packaged into a single, 25K byte program (passive state). But this version omitted all commentary and used multiple statements per line (at a saving of 5 bytes per statement line). Restoring commentary and making the programs more readable by stringing out each statement to a different line (with exceptions) ran the byte requirement up beyond the net core available in my microcomputer. This situation presented a small problem: How to have the best of both worlds. The solution was program linking (see *Personal Computing*, March '78, pp. 22-23). I broke the larger system into programs, none of which imposed a burden on computer core memory. As indicated by flowchart (see Figure 1), the programs pivot around a ROUTINE SELECTION (CRAS-RS: PROGRAM 3) program. Each time ROUTINE SELECT calls a program, the program, runs and then loops back with a RUN "CRAS-RS" statement, thus bringing you back to ROUTINE SELECT. You may then select a new routine or terminate operations.

CRAS Conveniences

A number of conveniences were built into the system. For example, during transaction entry, when the video terminal asks CHECK OR DEPOSIT, the user may answer with a "C" (Check) or "D" (deposit). Further, he may respond with a "B" (for balance), in which case the video will show his current balance in the check register. The video then returns to the original question, CHECK OR DEPOSIT.

One of the most frequent sources of errors occurs during transcription of account numbers. Master Charge and American Express account numbers, for example, each run 15 digits, while VISA runs 13 digits. These account numbers must be shown on checks used for account payment; also, the account number should be entered in the check register.

For convenience and to reduce errors, account numbers may be entered as program statements (see lines 580-600 of Program #4 – CRAS-TE) so they can be automatically reproduced on statements and checks whenever the name to which they relate (e.g., VISA) is entered. The examples provided in CRAS-TE may be extended to any reasonable length without noticeable delay in program execution.

Another convenience is the ACCOUNT, COMMA, AMOUNT loop. After you input the basic data relating to the check or deposit, the computer asks for the account number the entry is to be charged to, and then the amount to be charged to that account. Often you may want to distribute a check or deposit amount over more than one account. If you want to charge the entire amount of the check or deposit to a single account, merely enter a "0" (zero) after the account number, separating the two entries by a comma. The computer then charges the full value of the transaction to the specified account.

This program does more than balance your check book. It gives you a comprehensive accounting system for home or small business – or both, if you're self-employed.

You may want to distribute the amount of a check or deposit over several accounts – for example, when entering a list of items on a VISA or Master Charge statement. One item may relate to equipment purchased, another to office supplies and yet another to auto licenses (yes, you can even charge auto licenses here in Boulder County, Colorado). Further, you may have more than one entry for each of these accounts.

CRAS handles this problem easily. After you input the initial check data, you'll enter the ACCOUNT, COMMA, AMOUNT loop. Here, you may enter each item separately, charging it to the appropriate account. The computer acknowledges each entry with a formatted statement supplying the account number, the amount just charged to that account during the transaction entry, the subtotal of amounts charged to that account during the transaction entry and the balance remaining to be charged off. This loop continues until the amount you charge off to various accounts exactly equals the total amount of the transaction. When you want to charge all the balance remaining to a particular account, merely enter a "0" (zero) after the account number as the last entry. This feature is particularly convenient for charging off numerous miscellaneous account items.

When you make only a partial payment on the account or when you pay off a balance continued over from prior statements, you may want to charge off all of the items on an account statement even though the total may not equal the total of the transaction entry. If the total of account entries is less than the transaction amount (the amount of the check or deposit), merely charge the difference to a miscellaneous, catch-all account.

An entirely different problem occurs when the total of account entries exceeds the transaction amount. At the point in the ACCOUNT, COMMA, AMOUNT loop where the transaction amount is exceeded, the computer prints an advisory on the video terminal which states, THE TOTAL OF YOUR ENTRIES EXCEEDS THE AMOUNT OF THE CHECK OR DEPOSIT. DO YOU WISH TO ALLOW IT? In other words, you're given the option of overriding the algorithm. The computer then asks, DO YOU WISH TO REMAIN IN THE ACCOUNT, AMOUNT LOOP? This message assumes there will be times when you wish to charge off a number of items to particular accounts, even though each has exceeded the transaction amount.

CRAS also permits pro-rationing of the transaction amount between accounts on either a percentage or a frac-

tional basis – for example, if you operate a business from your home and desire to pro-rate utility expenses. In this instance you will want to pro-rate utilities between BUSINESS UTILITIES and PERSONAL UTILITIES accounts; or perhaps instead of PERSONAL UTILITIES, you'll charge MISCELLANEOUS. It's absurd to sit at a computer and make these calculations by pencil and paper – hence the automatic calculations in the ACCOUNT, COMMA, AMOUNT loop.

Simply input a 0, 0 (zero, comma, zero) when you enter the ACCOUNT, COMMA, AMOUNT loop, thus signaling the computer to enter the pro-rata loop. In this loop, the computer asks DENOMINATOR? If you are pro-rating on a percentage basis, the denominator is 100. If on a fractional basis, enter the denominator of the fraction. If, for example, one-sixth of your home is dedicated to business, enter a six.

The computer then asks, ACCOUNT NUMBER, COMMA, NUMERATOR? Enter the BUSINESS UTILITIES account number, a comma, then a "1", thus completing the fraction to be charged to that account. The computer applies this fraction to the transaction amount and charges the result against the account. If you desire to charge off the balance of the numerators to an account – in this case, 5 – you don't need to make the numerator calculation yourself; merely enter a "0" (zero) for the numerator amount, thus signaling the computer to pick up the balance of numerators, apply the resulting fraction to the transaction amount and charge the appropriate amount to the account specified.

You may want to charge off a transaction amount on a percentage basis between more than two accounts. OVERHEAD, for example, may be charged against five or six accounts. This situation poses no problem since the fractional ACCOUNT, COMMA, AMOUNT loop will cycle until the sum of the numerators equals the denominator. Should the sum of the numerators exceed the denominator, an advisory will be presented and the last entry may be over-written by the next input.

Typical checkbooks permit only brief notes explaining a transaction. We've all had trouble interpreting such brevity months later when reviewing our checkbooks at tax time.

CRAS recognizes the requirement for more flexible length when writing

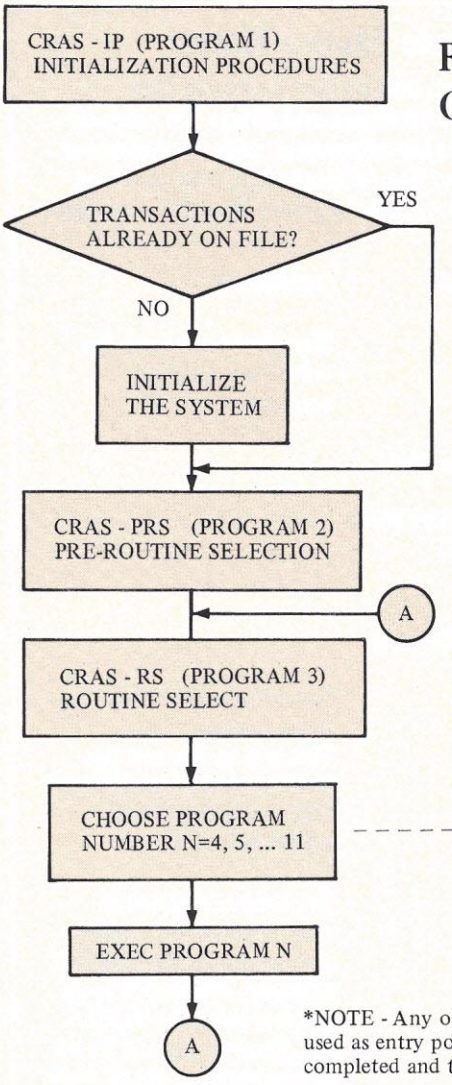


Figure 1
CRAS Flowchart

N	PROGRAM
4	CRAS - TE (PROGRAM 4) TRANSACTION ENTRY
5	CRAS - CR (PROGRAM 5) PRINT CHECK REGISTER STATEMENT
6	CRAS - BAD (PROGRAM 6) PRINT BUSINESS ACCOUNTS DISTRIBUTIONS
7	CRAS - PAD (PROGRAM 7) PRINT PERSONAL ACCOUNTS DISTRIBUTIONS
8	CRAS - SOA (PROGRAM 8) PRINT STATEMENTS OF ACCOUNTS
9	CRAS - CAR (PROGRAM 9) RECONCILE CHECKBOOK AND PRINT STATEMENT
10	CRAS - SF (PROGRAM 10) INPUT, DELETE, EDIT AND PRINT SUSPENSE FILE
11	CRAS - CP (PROGRAM 11) PRINT CHECKS

*NOTE - Any of the first three programs may be used as entry points after initialization has been completed and transactions are on file.

Table 1 – Memo Entries

DESIRED ADJUSTMENTS		ENTRIES		
Account to be Reduced	Account to be Increased	Check or Deposit	1st Entry	2nd Entry
Expense	Expense	Check	Minus	Plus
Expense	Income	Deposit	Plus	Plus*
Income	Expense	Check	Plus	Plus*
Income	Income	Deposit	Minus	Plus

***Where two "plus" entries are made, the advisory** YOU HAVE EXCEEDED THE TRANSACTION AMOUNT **will be made. An affirmative answer to the question,** DO YOU WISH TO ALLOW IT? **and** DO YOU WISH TO STAY IN THE ACCOUNT, COMMA, AMOUNT LOOP? **overides the advisory.**

Table 2 – Core Memory Requirements *

	Program	Core Required (bytes)	Function
#1	CRAS - IP	5099	Initialization Procedures
#2	CRAS - PRS	3339	Preliminary Routine Selection
#3	CRAS - RS	1912	Routine Selection
#4	CRAS - TE	7815	Transaction Entry
#5	CRAS - CR	7120	Check Register Statement

* Passive State

Figure 2 – Sample Memo Entry

Problem You wish to pick up savings interest for the quarter just passed on ACCOUNT #34 (Savings), and charge the catch-all account, ACCOUNT #40 (Miscellaneous), with the difference. The interest earned has not been deposited to your checking account, and so you do not want the Check Register disturbed by the transaction.

PROMPT	*RESPONSE*	*COMMENT*
CHECK OR DEPOSIT?	C	Since both accounts involved are expense accounts.
PAYEE/PAYOR	MEMO ENTRY	
IN PAYMENT OF	PICK UP SAVINGS INTEREST	
AMOUNT PAID	0	
AMOUNT OF CHECK	(Carriage Return)	
ACCOUNT, COMMA, AMOUNT	40, -23.45	Notice that the minus value was entered first. This was to prevent the YOU HAVE EXCEEDED THE TRANSACTION AMOUNT advisory.
ACCOUNT, COMMA, AMOUNT	34, 23.45	The loop continued to cycle because the amount charged in the ACCOUNT, COMMA, AMOUNT loop was less than the transaction amount (minus 23.45 as compared to zero). Now the amount charged to accounts (minus 23.45 and plus 23.45) equals the transaction amount – zero – and so we exit the loop.
DO YOU WISH TO ENTER A NOTE	Y	If you do not wish to enter a note, merely give a carriage return in response.
PLEASE INPUT THE TEXT OF YOUR NOTE	RESERVE FOR MORTGAGE INTEREST CHECKING ACCOUNT.	

Figure 3 – Transaction Entry Routine

PROMPT	*RESPONSE*
CHECK OR DEPOSIT?	"C" for check; "D" for deposit; or "B" for balance. "DONE" indicates you're through with the routine. This command may *not* be abbreviated.
DAY OF THE MONTH	1 or 2 digits; days need not be in order or consecutive (i.e., out of order days may be entered in the same month).

transaction notes. Hence, at the end of the TRANSACTION ENTRY loop, the computer will ask, DO YOU WISH TO ENTER A NOTE? If you answer yes, the computer will then instruct, PLEASE INPUT THE TEXT OF YOUR NOTE. You may then input up to 128 characters – a little over two ordinary typewritten lines of text.

Notes are printed as a separate statement following production of the Check Register; however, they are keyed to the transaction entry to which they relate by transaction number. This same keying scheme is used throughout the system with respect to all statements produced.

When you reach the end of the transaction entry, you're asked to PLEASE VERIFY ALL ENTRIES ON THIS TRANSACTION. Then, ARE THEY CORRECT? If you answer that they are correct ("Y"), the data is filed. But if you answer that they are not correct ("N"), all data entered in the transaction is erased, and you begin again with the same transaction number.

YES or NO responses may be abreviated to Y or N. Further, if the response is routine, you may eliminate it, merely hitting carriage return. Such questions such as DO YOU ALREADY HAVE TRANSACTIONS ON FILE recognize that the usual response will be YES; thus a carriage return will suffice for that reply. On the other hand, DO YOU WISH TO ENTER A NOTE will normally be answered NO; hence a carriage return is sufficient for that reply.

Memorandum Entries

MEMO ENTRIES (not to be confused with Transaction Memoranda, discussed elsewhere) provide another convenience. In reviewing an accounts distribution statement you may find you made an error, charging a transaction to the wrong account; or during the course of the year, you may decide to open a new account title and to charge off items previously reported in another account to the newly opened account; or, you may have been notified of interest income earned in your savings account, but which you wish to pick up in CRAS in the INTEREST INCOME account in such a way that the balance of the Check Register is not disturbed; or at some point in the year you may decide to pick up beginning-of-the-year balances for selected accounts (e.g., mortgage balance).

In each of these instances, you can

enter the data by DATA ENTRY. Respond with MEMO ENTRY when the PAYEE/PAYOR prompt appears on the video, *and* respond with "0" (zero) as the amount of the transaction. The zero assures that the balance of the check register will not be disturbed by the transaction.

But you must follow certain rules for MEMO ENTRY. First, ask yourself these questions: Shall the entry be made with a CHECK entry or a DEPOSIT entry, when that choice appears on the video? Should any of the entries be made as negative values? Then refer to Table 1. (See Figure 2 for an example of MEMO ENTRY.)

When using Table 1, keep in mind that accounts #25, 43, 44 and 45 are income accounts. Others are expense accounts. Also, you don't need to prefix values with a "plus"; the plus is implied.

Transfer of file balances and titling

CRAS can accomodate as many as 50 transactions per month (on average) for a total of about 600 transactions per diskette. You must anticipate when (if at all) you need to transfer your file balances and titling files to a fresh diskette. (A routine for this purpose will be provided in the last article in this series.)

Diskettes are cheap enough that you don't need to take chances by crowding the limit. The changeover must occur after you enter the last transaction for the month preceding the changeover. Changeover at midmonth prevents entering subsequent transactions for that month.

One user, for example, averaged 75 transactions per month, smoothly transfering balances to a new diskette at the end of the eighth month. If you average, say, 100 transactions per month, you should transfer diskettes at the end of the sixth month.

Memory requirements

Table 2 indicates the core memory requirements of programs (in a passive state) described in this article. Requirements for the remaining six programs of the package will be set forth in subsequent issues. Assuming you use a MITS DISK BASIC Interpreter (4.0 or 4.1), you should add about 20K bytes to the figures in the table, together with another K or two (for active runs), to find the minimum configuration of the system to be used. Thirty-two K of memory should be sufficient in any case.

PAYEE? (if check) or PAYOR? (if deposit)	As many as 30 characters may be used in the response. Additional characters will be accepted, but truncated.
IN PAYMENT OF?	As many as 26 characters may be used in the response. Additional characters will be accepted, but truncated.
AMOUNT PAID? (if check) or AMOUNT RECEIVED? (if deposit)	Use no dollar signs or commas (as in thousands). If a whole number, the decimal point need not be used, nor zeros entered to the right of it (e.g., $234.00 would be entered as "234").
DO YOU WANT A CHECK PRINTED ON THIS TRANSACTION?	Enter a "N" (No), or "Y" (Yes). A "Y" entry assumes that you will use the Check Printer routine later on. The full word may be entered if desired.
CHECK NUMBER? (if check)	If the previous question was answered "Y", this question will be by-passed. Further, this question will not appear with a Deposit transaction.
ACCOUNT, COMMA, AMOUNT	Enter the account number, then a comma, then the amount to be charged to the account. If the amount to be charged is the same as the transaction amount, a "0" may be entered, thus avoiding redundant entries. If the subtotal of all ACCOUNT, COMMA, AMOUNT entries is less than the transaction amount, the loop will continue to cycle, exiting only when the two values are the same. Enter a "0, 0" to signal entry to the fractional loop.
WHAT IS THE DENOMINATOR? (If the fractional loop has been signalled)	If the transaction amount is to be allocated on a percentage basis, enter "100"; otherwise enter the denominator of the fraction by which the transaction amount will be allocated.
ACCOUNT, COMMA, NUMERATOR? (If the fractional loop has been signalled)	Enter the account number to which the fractional part of the transaction amount is to be charged, then a comma, then the numerator. If you want to charge all remaining fractional parts to an account, enter a "0" for the numerator.
DO YOU WISH TO ENTER A NOTE?	Respond with a "Y" (Yes) or "N" (No)
PLEASE INPUT THE TEXT OF YOUR NOTE (If a "Y" was entered in response to the previous question)	The note may extend to 128 characters, including spaces.
PLEASE VERIFY YOUR ENTRIES ON THIS TRANSACTION. ARE THEY CORRECT?	Enter a "Y" or "N". "N" erases all entries on the transaction, and lets you reenter using the same transaction number.

The system will operate in 28K bytes for less powerful interpreters, but certain adjustments need to be made in the program (e.g., elimination of ELSE statements).

An easier way to reduce core requirements is to strip away all commentary and all program index lines (lines which merely provide space between program statements). Then, exercising caution, you can be even more economical by consolidating statements to the same line. Be careful, however, not to disturb statements referenced elsewhere in the program (e.g., GOTO, ON X GOTO, GOSUB, ON X GOSUB).

You can economize on core requirements and also on disk space by eliminating the SUSPENSE FILE routine. This routine reserves 120 disk records and also requires CLEARing about 2000 for string space. The file is useful and you should retain it if you have sufficient core to support it.

Another way to economize is to compromise. Instead of reserving 10 SUSPENSE FILE records per month, you may find you need only 5 and dispense with the rest. This adjustment will be discussed in a subsequent article.

Computer environment

Programming is in MITS DISK BASIC (4.0 or 4.1). Hardware consists of a microcomputer with 36K core memory, one disk drive, video terminal, and printing terminal with 132-character width.

Two input/output ports are used on the computer, addressed as 16 and 18. Sixteen is used for the video terminal and 18 for the printer. Switching from one to the other is fully automated in programming.

Other CRAS programs

This article deals with the first five programs of the CRAS System. Six remaining programs (seven, if the disk transfer program is included) will be discussed in later articles.

The first two of these remaining programs (CRAS-BAD and CRAS-PAD) provide printed statements of accounts distributions – one for business accounts and one for personal accounts. These programs give only summary information, but are nonetheless useful for entry verification and tabular footings. For example, in one sweep of the eye you can see current balances in key accounts.

One transaction entry provides you with a wealth specialized of outputs, including printed check the itself.

CRAS-SOA, the third program, provides a printed statement for each account. The statement contains full details of transactions on the account, including any transaction notes. Note, however, that transaction notes may be truncated due to space limitations on the printed page.

The fourth program (CRAS-CAR) permits notation of transactions which have cleared the bank. You can then obtain a Checking Account Reconciliation Statement. The statement also lists uncleared transactions in summary form.

The fifth program (CRAS-SF) permits entry of suspense file items and printing of the annual suspense file. The file may be edited, entries deleted, or entries added at any time during the year.

CRAS-CP, the sixth program, prints checks. This program, too, has a number of conveniences built in. You'll normally use this program immediately after entering transactions for which you want to write checks. The program prints the check amount in alphanumerics according to the desired format. After the check is written, the check number is filed with the transaction entry on file #1. The program will not permit writing more than one check on the same transaction.

To prevent mistakes, the last program (CRAS-DT) is not a part of the CRAS options shown in Program 3 (Routine Select); the program must be called by name rather than by routine number. The program transfers fixed and variable data from an old disk to a new one.

Sample CRT Entries

```
HAVE YOU FILED YOUR ACCOUNT TITLES? N

OK, NOW YOU MAY INPUT YOUR ACCOUNT TITLES. THESE WILL BE
FILED, SO IT WILL NOT BE NECESSARY TO DO THIS AGAIN.

AS EACH NUMBER APPEARS, PLEASE ENTER THE NAME OF THE ACCOUNT
TO BE ASSOCIATED WITH THAT NUMBER.

PLEASE REMEMBER THAT ACCOUNT NUMBERS 25, AND 43 THROUGH 45
MAY ONLY BE USED FOR INCOME ACCOUNTS. ALL OTHERS ARE EXPENSE
ACCOUNTS.

* * USE NO COMMAS OR COLONS IN YOUR ACCOUNT TITLES.

ACCOUNT NO.  10
? ELECTRONIC EQUIPMENT

ACCOUNT NO.  11
? ELECTRONIC SUPPLIES

ACCOUNT NO.  12
?
```

Filling in account titles during initialization.

```
                * ROUTINE SELECTION *

WHAT ROUTINE DO YOU WANT TO RUN AT THIS TIME (OR '10' TO
  SEE THE LIST)? 1

            * TRANSACTION ENTRY ROUTINE *

WHAT IS THE NUMBER OF THE MONTH?

TRANSACTION NO.  1
     CHECK OR DEPOSIT? D
     DAY OF THE MONTH? 3
     PAYOR? ELECTRONICS MANUFACTURING
     IN PAYMENT OF? EMPLOYEE SERVICES
     AMOUNT RECEIVED? 1234.56
     ACCOUNT NUMBER, COMMA, AMOUNT? 43,0
          * * ACT: 43    AMT:      1,234.56
     DO YOU WISH TO ENTER A NOTE?
PLEASE VERIFY YOUR ENTRIES ON THIS TRANSACTION.
          * * ARE THEY CORRECT?
```

Transaction entry routine was called for and the first transaction entered.

CHECK REGISTER FOR JANUARY, 1978

JOHN OR MARY DOE (A123 B456) / FIRST NATIONAL BANK AT HELSPORT

TXN NO.	DA	*	C	CHK# NO.	PAYEE/PAYOR	IN PAYMENT OF	AMOUNT PAID	AMOUNT RECD	BALANCE	DISTRIBUTIONS BUSINESS	DISTRIBUTIONS PERSONAL	N *
OPENING BALANCES									1234.56	0.00	1234.56	
1	3				ELECTRONICS MANUFACTURING	EMPLOYEE SERVICES	0.00	1234.56	2469.12	0.00	1234.56	
2	3				VARIOUS	SHOP RECEIPTS	0.00	123.45	2592.57	123.45	0.00	
3	3		-		FIRST NATIONAL BANK	HOME MORTGAGE PAYMENT	302.56	0.00	2290.01	20.00-	282.56-	*
4	3		-		MASTER CHARGE	ACCT. #12345678-9876543	0.00	0.00	2290.01	0.00	0.00	
5	3		****		MASTER CHARGE	ACCT. #12345678-9876543	143.25	0.00	2146.76	114.83-	28.42-	*
6	4		****		CANCER FUND	CONTRIBUTION	10.00	0.00	2136.76	0.00	10.00-	
7	4		****		VISA	ACCT. #9876 543 210 123	345.23	0.00	1791.53	268.19-	77.04-	*
8	10				VARIOUS	SHOP RECEIPTS	0.00	420.33	2211.86	420.33	0.00	
9	12		****		PLUMBER & ROOTER	CLOGGED SEWER	45.60	0.00	2166.26	7.60-	38.00-	
10	14		-		MEMO ENTRY	PICK UP MORTGAGE BAL.	0.00	0.00	2166.26	0.00	0.00	*
11	14		-		MEMO ENTRY	PICK UP SAVINGS BAL.	0.00	0.00	2166.26	0.00	0.00	*
CLOSING BALANCES FOR JANUARY, 1978							846.64	1778.34	2166.26	133.16	2033.10	

CHECK REGISTER NOTES FOR JANUARY, 1978

JOHN OR MARY DOE (A123 B456) / FIRST NATIONAL BANK AT HELSPORT

TRANSACTION NO. 3 :
( ) MORTGAGE PAYMENT DEDUCTED FROM CHECKING ACCOUNT EACH MONTH. INTEREST IS PRO-RATED BETWEEN BUSINESS (1/6) AND PERSONAL (5/6).

TRANSACTION NO. 5 :
( ) HAD LUNCH WITH BURT ROBBINS TO TALK OVER PURCHASE OF SHOP.

TRANSACTION NO. 7 :
( ) BOUGHT A MULTIMETER. CONTINUED DISCUSSION WITH BURT AT DINNER.

TRANSACTION NO. 10 :
( ) BALANCE OF HOME MORTGAGE AS OF JANUARY 1, 1978.

TRANSACTION NO. 11 :
( ) SAVINGS BALANCE AS OF JANUARY 1, 1978.

JOHN OR MARY DOE (A123 B456) / FIRST NATIONAL BANK AT HELSPORT

* * SCHEDULE OF ACCOUNTS * *

BUSINESS		PERSONAL	
10 ELECTRONIC EQUIPMENT	20 UTILITIES	26 UTILITIES	36 AUTO INSURANCE
11 ELECTRONIC SUPPLIES	21 TELEPHONE	27 TELEPHONE	37 LIFE INSURANCE
12 TECHNICAL SERVICES	22 INSURANCE	28 HOME MAINTENANCE	38 HOME INSURANCE
13 TECHNICAL LITERATURE	23 TAXES & LICENSES	29 HOME IMPROVEMENTS	39 TAXES & LICENSES
14 ASSOCIATIONS	24 INTEREST EXPENSE	30 MED. DENT. & PHARM.	40 MISCELLANEOUS
15 TRAVEL EXPENSE	25 BUSINESS INCOME	31	41 MORTGAGE & DEBT
16 ENTERTAINMENT		32 CONTRIBUTIONS	42 MORTGAGE INTEREST
17 OFFICE EQUIPMENT		33 INVESTMENTS	43 GENERAL INCOME
18 OFFICE SUPPLIES		34 SAVINGS	44 DIVIDEND INCOME
19 CLERICAL SERVICES		35 INTEREST EXPENSE	45 INTEREST INCOME

Program 1 Notes

Although Program #3 is the pivotal program, the system may proceed through Program #1 each time it is run. Program #1 initializes the system. The first question asked, DO YOU ALREADY HAVE TRANSACTIONS ON FILE, could easily be answered within the program by consulting the files. The reason it's not is to permit you to reinitialize all or part of the initialization entries at any time during the year. If you answered "Y" Program #1 terminates after first calling for Program #2 to be run.

If CRAS has not been initialized, you must supply a number of entries. Most of these entries become self-centering labels on printed statements.

The first entry is the name of the checking account. You should include the name(s) of the checking account, the account number and the name of the bank – for example JOHN OR MARY DOE (1234-5678)/ FIRST NATIONAL BANK AT PEOPLETOWN.

The second query asks for the *number* of the first month of the check register, to be used in subsequent error detection algorithms. Note that the CRAS fiscal year is the calendar year. All of the inputs thus far become a permanent record in File #9, "BT-INIT1".

The next query, the most time consuming, requires the most planning before responding. Although 36 accounts are made available to the user, I recommended that, unless you're already experienced in the use of those accounts, you regard your account structure as tentative, and not use them all. Usually a user will change his mind during the year, adding new account titles, deleting others and modifying the title of some, at least until the system becomes stabilized. For these reasons you're well advised to spend a lot of time structuring your accounts and the titles for each.

Next, the computer supplies the number of the account and asks for its title. Do not be concerned with clerical errors at this point. After all titles have been input you'll have an opportunity to produce a printed listing and to edit the account titles file.

When the computer asks for opening balances, you can take one of two approaches. Normally, expense and income accounts are zero balanced at the end of a fiscal period, and the only opening balance to be entered is the checking account balance. Thus,

the computer asks WHAT IS THE OPENING BALANCE IN YOUR CHECKING ACCOUNT, and then DO YOU WANT ALL REMAINING ACCOUNTS SET TO ZERO. The conventional approach is to answer in the affirmative.

An alternative (and in my view more useful) approach is to input opening balances for selected accounts, setting those remaining to zero. For example, you may want to maintain continuing records of your savings account balance, mortgage and personal debt, equipment purchased or home improvements. When the query is made, simply answer no ("N"), and you'll have an opportunity to review each account as the title appears and to input the opening balance for that account.

You should consider negative balances for some accounts. An account for DEBT, for example, should have the beginning of the year balance entered as a minus value. Thus each charge to the account during the year will reduce the balance. On the other hand a CHECK (e.g., loan proceeds) deposited to the check register increases the level of debt. This procedure applies a minus value to the DEBT account, thus increasing the balance.

Note that if you have transactions already filed in CRAS, you should not reinitialize. As long as you do not redefine the initial month or enter the opening balance routines, you may safely reinitialize the year, name of the checking account and account titles. But do note, you have an opportunity to edit or to print the account titles each time you use CRAS. Should you accidentally enter the opening balances routine after transactions have been placed on file, you should escape by "CONTROL-C" and rerun the program.

The "9" in statement 510 and elsewhere may puzzle you. While there are 36 accounts, the numbering of these accounts begins with #10 through #45.

Subroutines are made up of files. Files #5 and #6 are used to store account balances at the beginning of the month. To find an ending balance for a particular month, consult the month plus one record.

File #8 stores the number of the last transaction, the number of the last month for which a transaction has been entered, the record number of the last memorandum filed and the record number of the last transaction number in a continuous stream of transaction numbers which have cleared the bank. This feature prevents redundant searches for cleared transactions.

File #9 stores the name of the checking account and the name of the bank, together with the number of the initial month and the year. Note that this data is RSET (right set) in the disk field. The nulls at the left end of the field are stripped away beginning with line #1040, and the title established in line #1080.

File #10, consisting of six records with six sub-records on each (36 sub-records), stores account titles. Note that the maximum length of an account title is 21 characters. Account titles, too, are RSET; I'll provide a routine in subsequent programs to strip away the nulls, if any, preceding each account title and to define the title.

```
10 '    ***************************************************************************************************************************
20 '    *                                                                                                                         *
30 '    *                          CRAS (PROGRAM 1): CHECK REGISTER ACCOUNTING SYSTEM - O. E. DIAL   1978                       *
40 '    *                                                                                                                         *
50 '    ***************************************************************************************************************************
60 '
70 CLEAR 1000
80 DEFINT I
90 DIM AC$(36), NA$(6), AB(39), G2$(39)
100 '
110 '   ************************************************
120 '   *
130 '   *       CRAS-IP: INITIALIZATION PROCEDURE
140 '   *
150 '   ************************************************
160 '
170 PRINT TAB(30)"* * * CAUTION * * *": PRINT
180 PRINT "IF YOU ALREADY HAVE TRANSACTIONS ON FILE, INITIALIZATION PROCEDURES SHOULD BE"
190 PRINT "ENTERED ONLY FOR PURPOSES OF CHANGES IN THE CHECKING ACCOUNT TITLE.": PRINT
200 PRINT "MAKE NO CHANGES IN THE 'INITIAL MONTH' OR 'OPENING BALANCES.": PRINT: PRINT
210 PRINT "YOU MAY BY-PASS INITIALIZATION PROCEDURES BY THE QUESTIONS ABOUT TO BE ASKED.": PRINT
220 I8=0: Y$="": INPUT "DO YOU ALREADY HAVE TRANSACTIONS ON FILE"; Y$: PRINT
230     IF LEFT$(Y$,1)<>"N" THEN 840
240 Y$="": INPUT "DO YOU WISH TO BY-PASS INITIALIZATION PROCEDURES"; Y$: PRINT
250     IF LEFT$(Y$,1)<>"N" THEN 840
260     PRINT: PRINT: PRINT: PRINT TAB(10) "* * INITIALIZATION PROCEDURE * *": PRINT: PRINT
270     GOSUB 990'      * TO GET INITIALIZATION DATA AS MAY BE AVAILABLE
280 Y$="":INPUT "HAVE YOU FILED THE NAME OF YOUR CHECKING ACCOUNT"; Y$: PRINT:
        IF LEFT$(Y$,1)<>"N" THEN 320
290     PRINT "PLEASE SUPPLY THE NAME AND NUMBER OF YOUR CHECKING ACCOUNT,"
300     PRINT "AND THE NAME OF YOUR BANK."
310     LINE INPUT B$: PRINT: I8=1
320 Y$="": INPUT "HAVE YOU FILED THE OPENING MONTH"; Y$: PRINT:
        IF LEFT$(Y$,1)<>"N" THEN 360
330     PRINT "WHAT IS THE NUMBER OF THE FIRST MONTH OF YOUR CHECK REGISTER--"
340     INPUT "THE FIRST MONTH FOR WHICH YOU WILL RECORD TRANSACTIONS"; IM: PRINT: I7=1: I8=1
350     IF IM<1 OR IM>12 THEN 330
360 Y$="": INPUT "HAVE YOU FILED THE YEAR"; Y$: PRINT
370     IF LEFT$(Y$,1)<>"N" THEN 400
380     INPUT "WHAT IS THE YEAR"; YR$: PRINT: I8=1
390     IF LEN(YR$)<4 THEN PRINT "FOUR DIGITS, PLEASE.": PRINT: GOTO 380
400 IF I8=1 THEN GOSUB 880 ELSE 420
410     PRINT "* THE NAME OF YOUR ACCOUNT, BANK, FIRST MONTH, AND YEAR ARE NOW ON FILE. *": PRINT
420 Y$="": INPUT "HAVE YOU FILED YOUR ACCOUNT TITLES"; Y$: PRINT:
        IF LEFT$(Y$,1)<>"N" THEN 620
430     PRINT "OK, NOW YOU MAY INPUT YOUR ACCOUNT TITLES.  THESE WILL BE"
440     PRINT "FILED, SO IT WILL NOT BE NECESSARY TO DO THIS AGAIN.": PRINT
450     PRINT "AS EACH NUMBER APPEARS, PLEASE ENTER THE NAME OF THE ACCOUNT"
460     PRINT "TO BE ASSOCIATED WITH THAT NUMBER.": PRINT
470     PRINT "PLEASE REMEMBER THAT ACCOUNT NUMBERS 25, AND 43 THROUGH 45"
480     PRINT "MAY ONLY BE USED FOR INCOME ACCOUNTS.  ALL OTHERS ARE EXPENSE"
490     PRINT "ACCOUNTS.": PRINT
500     PRINT "* * USE NO COMMAS OR COLONS IN YOUR ACCOUNT TITLES.": PRINT
510     L=9: OPEN "R", 10, "BT-INIT2"
520     FOR J = 1 TO 6
530             FOR K = 1 TO 6: L = L + 1
540                     PRINT "ACCOUNT NO. " L
```

```
550                     INPUT AC$(K)
560                     GOSUB 950: PRINT
570             NEXT K
580             PUT #10, J
590     NEXT J: PRINT
600     CLOSE 10
610     PRINT "* YOUR ACCOUNT TITLES ARE NOW ON FILE. *": PRINT
620 Y$="": INPUT "HAVE YOU FILED YOUR OPENING BALANCES"; Y$: PRINT:
        IF LEFT$(Y$,1)<>"N" THEN 830
630 INPUT "WHAT IS THE OPENING BALANCE IN YOUR CHECKING ACCOUNT"; AB(37): PRINT
640             AB(39)=AB(37)
650     Y$="": INPUT "DO YOU WANT ALL REMAINING ACCOUNTS SET TO ZERO"; Y$: PRINT
660             IF LEFT$(Y$,1)<>"N" THEN 720
670     PRINT"OK, PLEASE ENTER THE OPENING BALANCE FOR EACH ACCOUNT.": PRINT
680             FOR J = 1 TO 36
690                     PRINT TAB(5) "ACCOUNT NUMBER "J+9 "   $";: INPUT AB(J): PRINT
700             NEXT J: AB(38)=0
710     GOTO 790
720     FOR J = 1 TO 38
730             IF J = 37 THEN 750
740             AB(J)=0
750     NEXT J
760 IF I7=1 THEN 780
770     GOSUB 990
780 M=IM
790     GOSUB 1110: M=M+1
800     GOSUB 1110: M=M-1'      THIS PLACES OPENING BALANCES ON THE MONTH DESIGNATED, AND ALSO THE MONTH FOLLOWING.
810 '                           THE OPENING BALANCES ON THE MONTH FOLLOWING WILL BE ADJUSTED WITH EACH BATCH OF TRANSACTIONS.
820     PRINT "* OK, YOUR ACCOUNT BALANCES ARE NOW ON FILE *": PRINT
830 GOSUB 1270'         * TO PLACE OPENING MONTH ON FILE #8
840 RUN"CRAS-PRS"
850 '
860 '   * * * SUBROUTINES * * *
870 '
880 OPEN "R", 9, "BT-INIT1"
890 FIELD #9, 2 AS IM$, 4 AS RY$, 122 AS NM$
900 RSET IM$=MKI$(IM): RSET RY$=YR$: RSET NM$=B$
910 PUT #9, 1
920 CLOSE
930 RETURN
940 '
950 FIELD #10, 2 AS D1$, (K-1) * 21 AS D2$, 21 AS NA$(K)
960 RSET NA$(K)=AC$(K)
970 RETURN
980 '
990 OPEN "R", 9, "BT-INIT1"
1000 FIELD #9, 2 AS IM$, 4 AS RY$, 122 AS NM$
1010 GET #9, 1
1020 IM=CVI(IM$): YR$=RY$: B$=NM$
1030 CLOSE
1040 FOR J1=1 TO 126
1050    IF MID$(B$, J1, 1)=" " THEN 1070
1060    GOTO 1080
1070 NEXT J1
1080 B$=MID$(B$, J1)
1090 RETURN
1100 '
1110 OPEN "R", 5, "BT-ACAB1"
1120 FOR I = 1 TO 16
1130    FIELD #5, 64 AS I1$, (I-1) * 4 AS I2$, 4 AS G2$(I)
1140    RSET G2$(I)= MKS$(AB(I))
1150 NEXT I
1160 PUT #5, M
1170 CLOSE 5
1180 OPEN "R", 6, "BT-ACAB2"
1190 FOR I = 17 TO 39
1200    FIELD #6, 36 AS I1$, (I-17) * 4 AS I2$, 4 AS G2$(I)
1210    RSET G2$(I) = MKS$(AB(I))
1220 NEXT I
1230 PUT #6, M
1240 CLOSE 6
1250 RETURN
1260 '
1270 OPEN "R", 8, "BT-LMLT1"
1280 FIELD #8, 2 AS M1$, 2 AS Q1$, 2 AS QM$, 2 AS QJ$, 120 AS DU$
1290 RSET M1$=MKI$(M): RSET Q1$=MKI$(Q): RSET QM$=MKI$(QN): RSET QJ$=MKI$(QJ)
1300 PUT #8, 1
1310 CLOSE
1320 RETURN
OK
```

Program 2 Notes

Program #2, preliminary to the routine selection program, is cycled through once for each run to let you edit account titles or obtain a printed listing of account titles.

The program also tells you the number of the last transaction entered into CRAS and the name of the last month for which transactions were entered. Note that you can enter transactions daily or on an unscheduled basis throughout the month. The month's entries are not closed out until the next month's number is entered for one or more transactions. You may not then go back to enter additional transactions in a prior month.

You may be puzzled by the strange FOR-NEXT loops in lines 240-290. The "J+ " features merely permit printing account titles in four columns, the two on the left reserved for business account titles, and the two on the right reserved for personal account titles.

Again, subroutines consist largely of files and file activity. Note that nulls are stripped from titles in lines 680-710 (checking account title), and in lines 810-150 (account titles).

Note, too, the routine in lines 930-980, which switches ports and rings the terminal bell to alert you to position paper in the printer for statement printing.

Line 1000 establishes the length of titling information and the amount of tab required to center the title on the statement. Line 1010 tabulates the title the required amount and prints it. This subroutine and the one previously discussed appear without change in all remaining programs.

```
10 '    ********************************************************************************************************************************************
20 '    *                                                                                                                                          *
30 '    *                          CRAS (PROGRAM 2): CHECK REGISTER ACCOUNTING SYSTEM - O. E. DIAL   1978                                          *
40 '    *                                                                                                                                          *
50 '    ********************************************************************************************************************************************
60 '
70 CLEAR 1000: WIDTH 132
80 DEFINT I-J: DEFINT M: DEFINT Q
90 DIM AC$(36)
100 '   ************************************************
110 '   *
120 '   *       CRAS-PRS: PRE-ROUTINE SELECTION
130 '   *
140 '   ************************************************
150 '
160 Y$="": INPUT "DO YOU WANT A LISTING OF THE ACCOUNT TITLES"; Y$: PRINT
170     IF LEFT$(Y$,1)<>"Y" THEN 320
180     GOSUB 630: I6=1'              * TO GET REPORT HEADING.  THE FLAG INDICATES THAT THE HEADING HAS BEEN OBTAINED.
190     GOSUB 750: I7=1'        * TO GET ACCOUNT TITLES.  THE FLAG INDICATES THAT THE ACCOUNT TITLES HAVE BEEN OBTAINED.
200     GOSUB 930'      * TO CHANGE CONSOLES FOR PRINTING
210     GOSUB 1000'     * TO POSITION AND PRINT TITLING DATA
220     PRINT TAB(51) "* * SCHEDULE OF ACCOUNTS * *": PRINT
230     PRINT" ----------------------BUSINESS-------------------------            ----------------------PERSONAL--------------------

---": PRINT
240     FOR J = 1 TO 6
250             PRINT J+9; AC$(J) TAB(33) J+19; AC$(J+10) TAB(66) J+25; AC$(J+16) TAB(99) J+35; AC$(J+26)
260     NEXT J
270     FOR J = 7 TO 10
280             PRINT J+9; AC$(J) TAB(66) J+25; AC$(J+16) TAB(99) J+35; AC$(J+26)
290     NEXT J
300     PRINTCHR$(12)'          * TO POSITION PAPER AFTER PRINTING
310     CONSOLE 16,0
320 Y$="": INPUT "DO YOU WANT TO EDIT THE ACCOUNT TITLES FILE"; Y$: PRINT
330     IF LEFT$(Y$,1)<>"Y" THEN 540
340     IF I7=1 THEN 360
350             GOSUB 750: I7=1'        * TO READ THE ACCOUNT TITLES
360     FOR J = 1 TO 36
370             PRINT "ACCOUNT NO. "J+9
380             PRINT AC$(J)
390             Y$="":INPUT "OK"; Y$: PRINT
400             IF LEFT$(Y$,1)<>"N" THEN 430
410                     PRINT "PLEASE TYPE THE NEW TITLE."
420                     LINE INPUT AC$(J): PRINT
430 NEXT J
440 OPEN "R", 10, "BT-INIT2"
450     FOR J = 1 TO 6
332             FOR K = 1 TO 6
470                     AC$(K)=AC$(K-6+(J*6))
352                     GOSUB 590'      * TO PUT THE REVISED ACCOUNT TITLES ON FILE
490             NEXT K
500             PUT #10, J
510     NEXT J: CLOSE
520     IF I6=1 AND I7=1 THEN 200
530     IF I6=1 THEN 190 ELSE 180
540 PRINT CHR$(12)'     * TO BLANK THE SCREEN
550 RUN "CRAS-RS"
560 '
570 '           * * * SUBROUTINES * * *
580 '
590 FIELD #10, 2 AS D1$, (K-1) * 21 AS D2$, 21 AS NA$(K)
600 RSET NA$(K)=AC$(K)
610 RETURN
620 '
630 OPEN "R", 9, "BT-INIT1"
640 FIELD #9, 2 AS IM$, 4 AS RY$, 122 AS NM$
650 GET #9, 1
660 IM=CVI(IM$): YR$=RY$: B$=NM$
670 CLOSE
680 FOR J1=1 TO 126
690     IF MID$(B$, J1, 1)=" " THEN 710
700     GOTO 720
710 NEXT J1
720 B$=MID$(B$, J1)
730 RETURN
740 '
750 OPEN "R", 10, "BT-INIT2"
760 FOR J = 1 TO 6
770     FOR K = 1 TO 6
780             FIELD #10, 2 AS D1$, (K-1) * 21 AS D2$, 21 AS AD$(K)
790     NEXT K
800     GET #10, J
810     FOR K = 1 TO 6
820             FOR J1 = 1 TO 21
830                     IF MID$(AD$(K), J1, 1)=" " THEN 850
840                     GOTO 860
850             NEXT J1
860             AC$(K-6+(J*6))=MID$(AD$(K), J1)
870     NEXT K
880 NEXT J
890 CLOSE
900 RETURN
910 '
920 '
930 CONSOLE 18,0
940     PRINT CHR$(7); CHR$(7); CHR$(7); CHR$(12)'      * TO RING THE BELL AND POSITION PAPER ON PRINTER.
950     CONSOLE 16,0
960     INPUT "PAPER POSITIONED"; Y$
970     CONSOLE 18,0
980 RETURN
990 '
1000 S=INT((132-LEN(B$))/2)
1010 PRINT TAB(S) B$: PRINT
1020 RETURN
OK
```

Program 3 Notes

Program #3 is the shortest but most important program in CRAS. This pivotal program asks you for the number of the routine to be run. Then, with an ON Y GOTO statement (line 300), the program directs the computer to run the proper program.

Note the parenthetical expression in line 160. As you gain experience using CRAS, you will not want to take time to review all of the numbered options; you'll know the number of the option you want from repeated prior use. But, for inexperienced users, option "10" causes the entire list of options and their numbers to be displayed.

The DON'T FORGET TO UNLOAD admonition in line 410 was added from experience. Before removing the CRAS disk from the MITS ALTAIR, the UNLOAD command must be given to prevent disk damage. This command ensures the print head is unloaded and the disk disabled. Failure to unload invites the egregious I/O ERROR advisory. Then you'll face the difficult problem of obtaining your file data.

So *don't forget to unload!*

```
10 '    ********************************************************************************************************************************
20 '    *                                                                                                                              *
30 '    *                    CRAS (PROGRAM 3): CHECK REGISTER ACCOUNTING SYSTEM - O. E. DIAL   1978                                    *
40 '    *                                                                                                                              *
50 '    ********************************************************************************************************************************
60 '
70 '
80 '    ************************************************
90 '    *
100 '   *       CRAS-RS: ROUTINE SELECTION
110 '   *
120 '   ************************************************
130 '
140 DEFINT Y
150 PRINT TAB(15) "* ROUTINE SELECTION *": PRINT
160 INPUT "WHAT ROUTINE DO YOU WANT TO RUN AT THIS TIME (OR '10' TO SEE THE LIST)"; Y: PRINT
170 IF Y < 10 THEN 300 ELSE IF Y>10 THEN 160 ELSE IF Y=10 THEN 180
180 PRINT TAB(3) "1. CREATE OR UPDATE THE CHECK REGISTER."
190 PRINT TAB(3) "2. PRINT THE CHECK REGISTER FOR THE MONTH TO BE INDICATED."
200 PRINT TAB(3) "3. PRINT THE BUSINESS ACCOUNT DISTRIBUTION STATEMENT FOR"
210 PRINT "THE MONTH TO BE INDICATED."
220 PRINT TAB(3) "4. PRINT THE PERSONAL ACCOUNT DISTRIBUTION STATEMENT FOR"
230 PRINT "THE MONTH TO BE INDICATED."
240 PRINT TAB(3) "5. PRINT SELECTED ACCOUNT STATEMENTS FOR THE PERIODS TO BE"
250 PRINT "INDICATED."
260 PRINT TAB(3) "6. CHECKBOOK RECONCILIATION"
270 PRINT TAB(3) "7. PRINT OR UPDATE THE SUSPENSE FILE."
280 PRINT TAB(3) "8. PRINT CHECKS
290 PRINT TAB(3) "9. END OPERATIONS.": PRINT
300 ON Y GOTO 320, 330, 340, 350, 360, 370, 380, 390, 410
310 '
320 RUN "CRAS-TE"
330 RUN "CRAS-CR"
340 RUN "CRAS-BAD"
350 RUN "CRAS-PAD"
360 RUN "CRAS-SOA"
370 RUN "CRAS-CAR"
380 RUN "CRAS-SF"
390 RUN "CRAS-CP"
400 '
410 PRINT"DONT FORGET TO UNLOAD! ! ! !": END
OK
```

Program 4 Notes

Study carefully Program #4, the Transaction Entry Routine, to discover its conveniences and flexibility; you'll spend more time with this program than any other.

After the number of the month is input, the program checks that the month hasn't been closed out by transactions in a later month. It then reads account balances in the month-plus-one record of files #5 and #6 (Lines 400-410).

Transaction entries begin by announcing the value "Q", the transaction number (Line 470). Input data as show in Figure 3.

Note that the last few entries are in the ACCOUNT, COMMA, AMOUNT loop. There are two such loops – one for entering absolute values (Lines 700-880) and one for entering fractions or percentages of the transaction value (Lines 890-960). If the sum of the entries exceeds the transaction amount in the absolute loop, an advisory is printed on the video terminal and you're given the option of continuing or reentering the transaction. If the sum of the numerators exceeds the denominator in the fractional loop, you're asked to reenter the transaction. All previous entries for that transaction are erased.

Lines 1040-1240 automatically adjust for pluses and minuses (distinguishing between income and expense accounts) according to whether the transaction is a check or deposit. Account entries are set and balances updated. In lines 1310-1320, the data is filed. The program then loops back to line 470.

In line 510, CHECK OR DEPOSIT?, a "DONE" response moves the program to line 2210. Lines 2220-2230 file the ending balances, and line 2240 files the last transaction number for the month. An advisory, ALL TRANSACTIONS ARE NOW FILED ON DISK is printed in line 2260 and in line 2280 the computer is instructed to run the ROUTINE SELECT program, CRAS-RS.

Subroutines, beginning on line 1360, are largely concerned with file activity. File #1 files transaction data by transaction number; #2 and #3 file

account entry data by transaction number; #4 files notes associated with particular transactions by note number. The note number (QN) is filed in file #1. Files #5 and 6 are account balance files, filed according to month, with the *opening* balance for the month appearing on each. File #7 records by month the last transaction number used for the month. File #8, a single-record file, records the last month and last transaction number for which data has been supplied.

The remaining subroutine in lines 2320-2350 opens all files for reading. Recall that many files can be open during GET routines, but that only one at a time may be open for PUT routines.

Notice the test in lines 790-800 and elsewhere. When working with floating point numbers, trailing, otherwise insignificant digits are often picked up because of imprecision in floating point arithmetic. Thus, a number such as "74.03" may become "74.030001". While this error in no way contributes to inaccuracy of the check register (since we are not interested in that level of precision), it does cause problems for equality tests, (e.g., IF A=B THEN . . .) Obviously 74.03 does not equal 74.030001; hence it would fail the test. Therefore, in the ACCOUNT, COMMA, AMOUNT loop, where this test is required, the precision tolerance has to be specified. In this case, AT (subtotal of all amounts entered) was set at ± (plus or minus) 9E-03, a sufficient tolerance (.009) for comparing the value with the transaction value.

Two codes are employed in filing transaction data. Both are an asterisk (Lines 650 (K6$) and 1010 (X$). These codes indicate whether a check is to be printed, and whether a note is associated with the transaction.

If you want to adjust the number of income accounts (versus the number of expense accounts), you'll need to change lines 1200-1230. For example, increasing the number of business income accounts from one (AE(16)) to two (AE(15) and AE(16) – accounts 24 and 25) requires adding line 1221 (AE(15)=AE(15)*(−1)). Further, you must change the terminal index of line 1200 from 15 to 14. Changing accounts on the "Personal" side requires similar adjustments to line 1210.

You can expand the total number of accounts within reason; the limit is governed solely by the number of columns which can be physically represented on one or more sheets of paper. In this instance, all account distributions are printed on two sheets. If you don't want printed distributions, you can easily expand the number of accounts to 100 or more. To keep printed distributions and still expand the number of accounts, you'll need to alter the print routines for the Accounts Distributions Programs.

```
10 '    ********************************************************************************************************************************
20 '    *                                                                                                                              *
30 '    *                        CRAS (PROGRAM 4): CHECK REGISTER BASED ACCOUNTING SYSTEM - O. E. DIAL   1978                         *
40 '    *                                                                                                                              *
50 '    ********************************************************************************************************************************
60 '
70 CLEAR 2500
80 DEFINT I-T: DEFINT E
90 '
100 W8$="ACT: ##    AMT:##,###.##   CUM:##,###.##   BAL:##,###.##"
110 W9$="ACT: ##     AMT: ####,###.##"
120 DATA JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER'     * LOAD THE CALENDAR
130 DIM M$(12), AB(39), AE(39), G1$(39), G2$(39)
140 FOR I=1 TO 12'      * READ THE CALENDAR
150     READ M$(I)
160 NEXT I
170 '
180 '
190 '   ************************************************
200 '   *
210 '   *       CRAS-TE: TRANSACTION ENTRY ROUTINE
220 '   *
230 '   ************************************************
240 '
250 GOSUB 2320' * TO OPEN THE FILES
260 GOSUB 1830' * TO GET LAST MONTH NUMBER AND TRANSACTION NUMBER
270 PRINT CHR$(12): PRINT TAB(10) "* TRANSACTION ENTRY ROUTINE *": PRINT
280     IF Q1=0 THEN 300
290 PRINT "YOUR LAST ENTRY WAS TRANSACTION " Q1 " IN " M$(M1) ".": PRINT
300 INPUT "WHAT IS THE NUMBER OF THE MONTH"; M: PRINT
310     IF M <1 OR M>12 THEN 300
320     IF M=M1 OR M=M1+1 THEN 400
330     IF M>M1+1 THEN PRINT "YOU ARE SKIPPING A MONTH.  PLEASE REENTER THE MONTH NUMBER.": PRINT: GOTO 300
340 IF M => M1 THEN 400
350     PRINT "TRANSACTIONS FOR THE MONTH OF " M$(M)" HAVE BEEN CLOSED OUT"
360 PRINT "BY TRANSACTIONS ENTERED IN " M$(M1)".   YOU MAY ENTER YOUR NEW"
370 PRINT "TRANSACTIONS IN " M$(M1) " OR LATER.": PRINT
380 PRINT "PLEASE REENTER THE MONTH.": PRINT
390 GOTO 300
400 M=M+1
410 GOSUB 1680: M=M-1'  * TO GET OPENING BALANCES
420 Q=Q1: QN=QM
430 CLOSE
440 '
450 '           * * * TRANSACTION ENTRIES * * *
460 '
470 Q=Q+1
480     A1$="": A2$="": X$="": A4$=" ": A5$="": A6$="": A7$="": A8$="": A1=0: K6$=""
490     PRINT CHR$(12)' * TO BLANK THE SCREEN
500     PRINT "TRANSACTION NO. " Q
510     PRINT TAB(5);: INPUT "CHECK OR DEPOSIT"; A2$
520     IF LEFT$(A2$,1)="B" THEN PRINT: PRINT "* * THE CHECK REGISTER BALANCE IS $"AB(37): PRINT: GOTO 510
530     IF A2$="DONE"  THEN Q=Q-1: GOTO 2210
540     A2$=LEFT$(A2$,1)
550     IF A2$<>"C" AND A2$<>"D" THEN 510
560     PRINT TAB(5);: INPUT "DAY OF THE MONTH";A1$
570     PRINT TAB(5);: IF A2$="C" THEN INPUT "PAYEE"; A5$ ELSE INPUT "PAYOR"; A5$
580     IF A5$="MASTER CHARGE" THEN A6$="ACCT. #12345678-9876543": GOTO 620
590     IF A5$="VISA" THEN A6$="ACCT. #9876 543 210 123": GOTO 620
600     IF A5$="AMERICAN EXPRESS" THEN A6$="ACCT. #9999-888888-77777": GOTO 620
610     PRINT TAB(5);: LINE INPUT "IN PAYMENT OF?"; A6$
620     PRINT TAB(5);: IF A2$="C" THEN INPUT "AMOUNT PAID"; A1 ELSE INPUT "AMOUNT RECEIVED"; A1
630     IF A2$="D" THEN 680
640     Y$="": PRINT TAB(5);: INPUT "DO YOU WANT A CHECK PRINTED ON THIS TRANSACTION"; Y$
650             IF LEFT$(Y$,1)="Y" THEN K6$="*": A7$="*****": GOTO 680
660     PRINT TAB(5);: INPUT "CHECK NUMBER"; A7$
```

```
670             IF A7$="" THEN A7$=" - "
680     AT=0: AU=0: A2=0: R=0: ET=0: ED=0: EN=0
690     FOR J=1 TO 39: AE(J)=0: NEXT J
700     PRINT TAB(5);:INPUT "ACCOUNT NUMBER, COMMA, AMOUNT"; R, A2: R=R-9:IF R>36 THEN PRINT "ERRONIOUS ENTRY": GOTO 700
710             IF R=(-9) THEN 890
720             IF R<1 THEN 700
730             IF A2<>0 THEN 760
740             PRINT TAB(10)"* * ";: PRINT USING W9$; R+9,AE(R)+ A1-AT
750             AE(R)=AE(R)+A1-AT : GOTO 970
760             AE(R)=AE(R)+A2
770             AT=AT+A2: AU=A1-AT
780             PRINT "* ";: PRINT USING W8$;R+9,AE(R),AT,AU: PRINT
790             IF AT<A1-9E-03 THEN 700
800             IF AT< A1+9E-03 THEN 970
810             PRINT "THE TOTAL OF YOUR ENTRIES EXCEEDS THE AMOUNT OF THE CHECK OR DEPOSIT."
820                     Y$="": INPUT "DO YOU WISH TO ALLOW IT"; Y$: PRINT
830                     IF LEFT$(Y$,1)<>"Y"THEN 870
840                     Z$="": INPUT "DO YOU WISH TO REMAIN IN THE 'ACCOUNT, AMOUNT' LOOP";Z$: PRINT
850                             IF LEFT$(Z$,1)="Y" THEN 700
860                             IF LEFT$(Y$,1)="Y" THEN 970
870                     PRINT "THE ACCOUNT/AMOUNT ENTRIES FOR THIS TRANSACTION HAVE BEEN REVERSED."
880                     PRINT "PLEASE REENTER THE TRANSACTION.": PRINT: Q=Q-1: GOTO 470
890             INPUT "WHAT IS THE DENOMINATOR"; ED
900             INPUT "ACCOUNT, COMMA, NUMERATOR"; R, EN: R=R-9:
                        IF R<1 THEN 900 ELSE IF R>36 THEN PRINT "ERRONIOUS ENTRY.": GOTO 900
910             IF EN=0 THEN AE(R)=A1-AT: AT=A1: ET=ED: GOTO 950
920             AE(R)=EN*A1/ED
930             ET=ET+EN: AT=AT+AE(R)
940             IF ET>ED THEN ET=0: AT=0: FOR J = 1 TO 36: AE(J)=0: NEXT J:
                        PRINT "THE TOTAL OF YOUR NUMERATORS EXCEEDS THE DENOMINATOR.":
                        PRINT "PLEASE BEGIN AGAIN.": GOTO 890
950             PRINT  "* ";: PRINT USING W8$; R+9,AE(R),AT,A1-AT
960             IF ET<ED-9E-03 THEN 900
970     Y$="": PRINT TAB(5);:LINE INPUT "DO YOU WISH TO ENTER A NOTE?"; Y$
980     IF LEFT$(Y$,1)<>"Y" THEN 1020
990             PRINT "PLEASE INPUT THE TEXT OF YOUR NOTE."
1000            LINE INPUT A8$
1010            X$="*": QN=QN+1
1020    PRINT "PLEASE VERIFY YOUR ENTRIES ON THIS TRANSACTION."
1030    Y$="":PRINT TAB(10) "* * ";: INPUT "ARE THEY CORRECT"; Y$
1040            IF LEFT$(Y$,1)<>"N" THEN 1090
1050            PRINT "OK, ALL ENTRIES WITH RESPECT TO THIS TRANSACTION HAVE BEEN REVERSED."
1060            PRINT "PLEASE REENTER TRANSACTION"Q".": PRINT
1070            IF X$="*" THEN QN=QN-1
1080            GOTO 480
1090 FOR K = 1 TO 16
1100    A3=AE(K)
1110    IF A2$="C" THEN A3=A3*(-1)
1120    AE(38)=AE(38)+A3:
        AB(38)=AB(38)+A3
1130 NEXT K
1140 FOR K = 17 TO 36
1150    A3=AE(K)
1160    IF A2$="C" THEN A3=A3*(-1)
1170    AE(39)=AE(39)+A3:
        AB(39)=AB(39)+A3
1180 NEXT K
1190 IF A2$<>"D" THEN 1220
1200 FOR K = 1 TO 15:
        AE(K)=AE(K)*(-1):
        NEXT K
1210 FOR K = 17 TO 33:
        AE(K)=AE(K)*(-1):
        NEXT K:
        GOTO 1240
1220 AE(16)=AE(16)*(-1)
1230 FOR K = 34 TO 36:
        AE(K)=AE(K)*(-1):
        NEXT K
1240 FOR K = 1 TO 36
1250    AB(K)=AB(K)+AE(K):
        NEXT K
1260 A3=A1
1270 IF A2$="C" THEN A3=A3*(-1)
1280 AB(37)=AB(37)+A3
1290 AE(37)=AB(37)
1300 '
1310 GOSUB  1360'       * TO PUT THE TRANSACTION ON FILE
1320 GOSUB 1430: GOTO 470'      * TO PUT THE ACCOUNT DATA ON FILE
1330 '
1340 '          * * * SUBROUTINES
1350 '
1360 OPEN "R", 1, "BT-TRAN1"
1370 FIELD #1,2 AS FM$, 2 AS F1$, 1 AS F2$, 1 AS FX$, 1 AS F4$, 30 AS F5$, 30 AS F6$, 4 AS F7$, 2 AS F9$, 4 AS FA$, 1 AS FB$, 50 AS

DU$
1380 RSET FM$=MKI$(M): RSET F1$=A1$: LSET F2$=A2$: LSET FX$=X$: LSET F4$=A4$: LSET F5$=A5$: LSET F6$=A6$:  RSET F7$=A7$: RSET F9$=MK

I$(QN): RSET FA$=MKS$(A1): LSET FB$=K6$
1390 PUT #1, Q
1400 CLOSE
1410 RETURN
1420 '
1430 OPEN "R", 2, "BT-ACEN1"
1440 FOR I=1 TO 16
1450    FIELD #2, 64 AS I1$, (I-1) * 4 AS I2$, 4 AS G1$(I)
1460    RSET G1$(I)=MKS$(AE(I))
1470 NEXT I
1480 PUT #2, Q
1490 CLOSE
1500 '
1510 OPEN "R", 3, "BT-ACEN2"
1520 FOR I=17 TO 39
1530    FIELD #3, 36 AS I1$, (I-17) * 4 AS I2$, 4 AS G1$(I)
1540    RSET G1$(I)=MKS$(AE(I))
1550 NEXT I
1560 PUT #3, Q
1570 CLOSE
1580 '
```

```
1590 IF X$<>"*" THEN 1660
1600 '
1610 OPEN "R", 4, "BT-NOTE1"
1620 FIELD #4, 128 AS F8$
1630 RSET F8$=A8$
1640 PUT #4, QN
1650 CLOSE
1660 RETURN
1670 '
1680 FOR I = 1 TO 16
1690    FIELD #5, 64 AS I1$, (I-1) * 4 AS I2$, 4 AS G2$(I)
1700 NEXT I
1710 GET #5, M
1720 '
1730 FOR I = 17 TO 39
1740    FIELD #6, 36 AS I1$, (I-17)*4 AS I2$, 4 AS  G2$(I)
1750 NEXT I
1760 GET #6, M
1770 '
1780 FOR I = 1 TO 39
1790    AB(I)=CVS(G2$(I))
1800 NEXT I
1810 RETURN
1820 '
1830 FIELD #8, 2 AS M1$, 2 AS Q1$, 2 AS QM$, 2 AS QJ$, 120 AS DU$
1840 GET#8, 1
1850 Q1=CVI(Q1$): M1=CVI(M1$): QM=CVI(QM$): QK=CVI(QJ$)
1860 RETURN
1870 '
1880 OPEN "R", 5, "BT-ACAB1"
1890 FOR I = 1 TO 16
1900    FIELD #5, 64 AS I1$, (I-1) * 4 AS I2$, 4 AS G2$(I)
1910    RSET G2$(I)= MKS$(AB(I))
1920 NEXT I
1930 PUT #5, M
1940 CLOSE
1950 '
1960 OPEN "R", 6, "BT-ACAB2"
1970 FOR I = 17 TO 39
1980    FIELD #6, 36 AS I1$, (I-17) * 4 AS I2$, 4 AS G2$(I)
1990    RSET G2$(I) = MKS$(AB(I))
2000 NEXT I
2010 PUT #6, M
2020 CLOSE
2030 RETURN
2040 '
2050 OPEN "R", 7, "BT-TMTN1"
2060 FIELD #7, 126 AS D$, 2 AS Q$
2070 RSET Q$=MKI$(Q)
2080 PUT #7, M
2090 CLOSE
2100 '
2110 QJ=QK
2120 OPEN "R", 8, "BT-LMLT1"
2130 FIELD #8, 2 AS M1$, 2 AS Q1$, 2 AS QM$,  2 AS QJ$, 120 AS DU$
2140 RSET M1$=MKI$(M): RSET Q1$=MKI$(Q): RSET QM$=MKI$(QN): RSET QJ$=MKI$(QJ)
2150 PUT #8, 1
2160 CLOSE
2170 RETURN
2180 '
2190 '          * * * PROGRAM CONTINUATION
2200 '
2210 IF Q=Q1 THEN 2270
2220 M=M+1
2230 GOSUB 1880: M=M-1' * TO PUT THE NEW BALANCES ON FILE
2240 GOSUB  2050'       * TO PUT THE LAST TRANSACTION NUMBER FOR THE MONTH ON FILE #7
2250 '                  * THEN TO PUT UPDATE ON FILE #8
2260 PRINT: PRINT "* * ALL TRANSACTIONS ARE NOW FILED ON DISK. * *": PRINT
2270 CLOSE
2280 RUN "CRAS-RS"
2290 '
2300 '          * * * SUBROUTINE CONTINUATION
2310 '
2320 OPEN "R", 5, "BT-ACAB1"
2330 OPEN "R", 6, "BT-ACAB2"
2340 OPEN "R", 8, "BT-LMLT1"
2350 RETURN
OK
```

Program 5 Notes

Program #5 prints the Check Register for the month selected, or at any time during the month for as many transactions as you've entered. For example, you can print interim Check Registers as working documents during the month, then replace these with a final Check Register at the end of the month.

For that matter, the "final" Check Register is not really final until all transactions for the month have cleared the bank. Note that the third column of the Check Register is captioned "C". This column contains a star for each transaction which has cleared the bank. The star is placed on file when the Checkbook Reconciliation Routine is run. Check Registers printed afterwards show the star in that column for cleared transactions. Therefore, you should print a fresh Check Register for months showing uncleared transactions each time you run the Checkbook Reconciliation Routine.

The first column of the Check Register shows Transaction Numbers (TXN), which run consecutively through the year. They constitute the record key for all printed documents.

All remaining Check Register headings are self-explaining, except for the last column, "*". A star in this column indicates there is a memo associated with the transaction. If a star appears in this column for any transaction on the Check Register, the program automatically prints a Check Register Memorandum Statement immediately following the Check Register Statement.

```
10 '    ***************************************************************************************************************************
20 '    *                                                                                                                         *
30 '    *                        CRAS (PROGRAM 5): CHECK REGISTER BASED ACCOUNTING SYSTEM - O. E. DIAL   1978                    *
40 '    *                                                                                                                         *
50 '    ***************************************************************************************************************************
60 '
70 CLEAR 1000: WIDTH 132
80 DEFINT I-T
90 '
100 '           * * PRINT USING FORMATS
110 '
120 W1$="###|\\|!|\  \|\                          \ | \                        \ |######.##-|######.##-|######.##-||######.##-|#####

#.##-|!|"
130 W2$="OPENING BALANCES                          |                            |          |          |######.##-||######.##-|#####

#.##-||"
140 W3$="|######.##-|######.##-|######.##-||######.##-|######.##-||"
150 '
160 DATA JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER'      * LOAD THE CALENDAR
170 DIM M$(12), AB(39), AE(39), G1$(39), G2$(39)
180 FOR I=1 TO 12'      * READ THE CALENDAR
190     READ M$(I)
200 NEXT I
210 GOTO 780
220 '
230 '   ************************************************
240 '   *
250 '   *       CRAS-CR: CHECK REGISTER STATEMENT
260 '   *
270 '   ************************************************
280 '
290 '           * * * SUPPORTING FILES
300 '
310 FIELD #9, 2 AS IM$, 4 AS RY$, 122 AS NM$
320 GET #9, 1
330 IM=CVI(IM$): YR$=RY$: B$=NM$
340 FOR J1=1 TO 126
350     IF MID$(B$, J1, 1)=" " THEN 370
360     GOTO 380
370 NEXT J1
380 B$=MID$(B$, J1)
390 RETURN
400 '
410 FIELD #7, 126 AS D$, 2 AS O$
420 GET #7, M
430 O = CVI(O$)
440 '
450 FIELD #8, 2 AS M1$, 2 AS O1$, 2 AS OM$, 2 AS OJ$, 120 AS DU$
460 GET #8, 1
470 O1=CVI(O1$): M1=CVI(M1$): OM=CVI(OM$): OK=CVI(OJ$)
480 RETURN
490 '
500 FIELD #1, 2 AS FM$, 2 AS F1$, 1 AS F2$, 1 AS FX$, 1 AS F4$, 30 AS F5$, 30 AS F6$, 4 AS F7$, 2 AS F9$, 4 AS FA$, 1 AS FB$, 50 AS

DU$
510 GET #1,  J
520 A1=CVS(FA$): QN=CVI(F9$): A1$=F1$: A7$=F7$: A2$=F2$: X$=FX$: A4$=F4$: A5$=F5$: A6$=F6$: K6$=FB$
530 RETURN
540 '
550 FOR I = 1 TO 16
560     FIELD #2, 64 AS I1$, (I-1) * 4 AS I2$, 4 AS G1$(I)
570 NEXT I
580 GET #2, J
590 FOR I = 17 TO 39
600     FIELD #3, 36 AS I1$, (I-17) * 4 AS I2$, 4 AS G1$(I)
610 NEXT I
620 GET #3, J
630 FOR I =1 TO 39
640     AE(I)=CVS(G1$(I))
650 NEXT I
660 RETURN
670 '
680 FIELD #4, 128 AS F8$
690 GET #4, QN: A8$=F8$
700 FOR I = 1 TO 128
710     IF MID$(A8$, I, 1)=" " THEN 730
720     A8$=MID$(A8$, I): GOTO 740
730 NEXT I
740 RETURN
750 '
760 '           * * * PROGRAM CONTINUATION
770 '
780 PRINT CHR$(12)'     TO BLANK THE SCREEN
790 PRINT TAB(10) "* CHECK REGISTER STATEMENT ROUTINE *": PRINT
800 INPUT "WHAT IS THE NUMBER OF THE MONTH"; M: PRINT
810 GOSUB 1830'         * TO OPEN THE FILES
820 GOSUB 1760'         * TO TEST FOR THE CORRECT MONTH
830 GOSUB 1930'         * TO FORM THE DATE LINE
840 GOSUB 1960'         * TO CHANGE CONSOLES FOR PRINTING THE STATEMENT
850 GOSUB 1370'         * TO PRINT DOUBLE UNDERSCORE
860 GOSUB 1430'         * TO ADD DATE LINE TO STATEMENT TITLE
870 GOSUB 1500'         * PRINT NAME OF CHECKING ACCOUNT
880 GOSUB 1400'         * PRINT DOUBLE UNDERSCORE
890 GOSUB 1540'         * PRINT COLUMN HEADINGS
900 GOSUB 920: GOTO 1010'       * TO GET ACCOUNT OPENING BALANCES
910 '
920 FOR I = 37 TO 39
930     FIELD #6, 36 AS I1$, (I-17)*4 AS I2$, 4 AS G2$(I)
940 NEXT I
950 GET #6, M
960 FOR I = 37 TO 39
970     AB(I)=CVS(G2$(I))
980 NEXT I
990 RETURN
1000 '
1010 PRINT USING W2$; AB(37), AB(38), AB(39)'   * PRINT THE OPENING BALANCES
1020 GOSUB 2030'        * GET THE OPENING AND CLOSING TRANSACTION NUMBERS FOR THE MONTH
1030 I9=0
1040 FT=0: GT=0'        * INITIALIZE THE ACCUMULATORS (BUSINESS AND PERSONAL ENTRIES, RESPECTIVELY)
```

Program 5 - Continued

```
1050 FOR J = P TO Q
1060    GOSUB 500'      * GET TRANSACTION DATA
1070    GOSUB 550'      * GET ACCOUNT DATA FOR THE TRANSACTION
1080    IF X$="*" THEN I9=1'    * TEST FOR TRANSACTION MEMORANDA AND SET MEMO FLAG
1090    GOSUB 1630'     * PRINT THE TRANSACTION
1100 NEXT J
1110 M=M+1
1120 GOSUB 920: M=M-1'  * GET CLOSING BALANCES
1130 GOSUB 1720'        * PRINT CLOSING BLANCES
1140 GOSUB 1370'        * PRINT DOUBLE UNDERSCORE
1150 IF I9<>1 THEN 1300'*TEST FOR MEMORANDA
1160 CONSOLE 16, 0
1170 GOSUB 1960'        * CHANGE CONSOLES TO PRINT MEMORANDA STATEMENT
1180 GOSUB 1370'        * PRINT DOUBLE UNDERSCORE
1190 GOSUB 1590'        * PRINT STATEMENT TITLE AND DATE LINE
1200 GOSUB 1500'        * PRINT NAME OF CHECKING ACCOUNT
1210 GOSUB 1400'        * PRINT SINGLE UNDERSCORE
1220 FOR J = P TO Q
1230    GOSUB 500'      * CHECK FLAG FOR MEMORANDUM
1240    IF X$<>"*" THEN 1280
1250    GOSUB 680'      * GET MEMORANDUM
1260    PRINT "TRANSACTION NO." J;":"
1270    PRINT"( ) "; A8$: PRINT'        * PRINT THE MEMORANDUM
1280 NEXT J: PRINT
1290 GOSUB 1370'        * PRINT DOUBLE UNDERSCORE
1300 PRINT CHR$(12): PRINT CHR$(12)'    * TO POSITION PAPER FOR REMOVAL OF STATEMENT
1310 CONSOLE 16,0
1320 CLOSE
1330 RUN "CRAS-RS"
1340 '
1350 '          * * * SUBROUTINES
1360 '
1370 FORJ2=1TO131:PRINT"=";:NEXTJ2:PRINT: PRINT'        * PROVIDES DOUBLE UNDERSCORE
1380 RETURN
1390 '
1400 FORJ2=1TO131:PRINT"-";:NEXTJ2:PRINT'               * PROVIDES SINGLE UNDERSCORE
1410 RETURN
1420 '
1430 Y$= "CHECK REGISTER FOR " + TD$'           * CONCATENATES THE  ELEMENTS OF THE TITLE
1440 GOSUB 1470
1450 RETURN
1460 '
1470 PRINT TAB(INT((132-LEN(Y$))/2)) Y$: PRINT' * CALCULATES THE TITLE TABULATION
1480 RETURN
1490 '
1500 S=INT((132-LEN(B$))/2)'    * CALCULATES THE TABULATION FOR THE REMAINING TITLE
1510 PRINT TAB(S) B$: PRINT
1520 RETURN
1530 '
1540 PRINT"TXN|DA|C|CHK#|" TAB(23) "PAYEE/PAYOR" TAB(43) "|" TAB(52) "IN PAYMENT OF" TAB(72)"|  AMOUNT  |  AMOUNT  | BALANCE  ||

DISTRIBUTIONS    ||N"
1550 PRINT"NO.|  |*| NO.|"TAB(43)"|"TAB(72)"|   PAID   |   RECD   |          || BUSINESS | PERSONAL ||*
1560 PRINT"---|--|-|----|------------------------------|------------------------------|----------|----------|----------||----------

-------||-"
1570 RETURN
1580 '
1590 Y$= "CHECK REGISTER NOTES FOR " + TD$'     * CONCATENATES THE ELEMENTS OF THE TITLE
1600 GOSUB 1470
1610 RETURN
1620 '
1630 IF A2$="C" THEN F1=A1 ELSE F2=A1
1640 IF A4$<>"*" THEN A4$=""
1650 IF X$<>"*" THEN X$=""
1660 PRINT USING W1$; J; A1$;A4$; A7$; A5$; A6$; F1; F2; AE(37); AE(38); AE(39); X$
1670 A1$="": A5$="": A6$="": A7$="": A4$="": X$=""
1680 FT=FT+F1: GT=GT+F2'        * ACCUMULATES THE 'CHECKS' AND 'DEPOSITS', RESPECTIVELY
1690 F1=0: F2=0
1700 RETURN
1710 '
1720 PRINTTAB(72)"|----------|----------|----------||----------|----------||"
1730 PRINT "CLOSING BALANCES FOR " M$(M)", "; YR$;: PRINT TAB(72);: PRINT USING W3$; FT; GT; AB(37); AB(38); AB(39)
1740 RETURN
1750 '
1760 GOSUB 450: GOSUB 310: IF M=IM THEN 1810'   * TESTS FOR ERRONIOUS MONTH ENTRY
1770 IF M<=M1 THEN 1810
1780 PRINT "THE LAST MONTH FOR WHICH YOU HAVE ENTERED TRANSACTIONS IS"
1790 PRINT M$(M1)".  WE HAVE SUBSTITUTED THIS AS THE MONTH OF THE STATEMENT"
1800 PRINT "YOU WANT.": PRINT: M=M1
1810 RETURN
1820 '
1830 OPEN "R", 1, "BT-TRAN1"
1840 OPEN "R", 2, "BT-ACEN1"
1850 OPEN "R", 3, "BT-ACEN2"
1860 OPEN "R", 4, "BT-NOTE1"
1870 OPEN "R", 6, "BT-ACAB2"
1880 OPEN "R", 7, "BT-TMTN1"
1890 OPEN"R", 8, "BT-LMLT1"
1900 OPEN "R", 9, "BT-INIT1"
1910 RETURN
1920 '
1930 TD$=M$(M) + ", " + YR$'    * CONCATENATES THE ELEMENTS OF THE DATE LINE
1940 RETURN
1950 '
1960 CONSOLE 18,0'      * PROVIDES PROMPT TO POSITION PAPER ON THE PRINTING TERMINAL
1970    PRINT CHR$(7); CHR$(7); CHR$(7); CHR$(12)'      * TO RING BELL AND POSITION PAPER
1980    CONSOLE 16,0
1990    INPUT "PAPER POSITIONED"; Y$: PRINT
2000    CONSOLE 18,0
2010 RETURN
2020 '
2030 IF M=IM THEN P=1: GOTO 2060'       * PICKS UP THE FIRST AND LAST TRANSACTION NUMBERS FOR THE MONTH
2040 M=M-1
2050 GOSUB 410: P=Q+1: M=M+1
2060 GOSUB 410
2070 RETURN
OK
```

Secrecy
and your Personal Computer

BY STEPHEN SMITH

In this age of electronic surveillance, are your personal and business communications secure?

Picture the situation of two East Coast contractors bidding on the same $5 million job in California. Both are in constant communication with their West Coast representatives by mail and telephone. Each would like to know what the other is planning, and, with $5 million at stake, would use almost any means to find out.

One contractor relies on the (limited) security of normal mail and wire communications. The other uses a microcomputer to encode and decode messages at his offices on each coast. Which contractor do you think is more likely to have his messages intercepted and read?

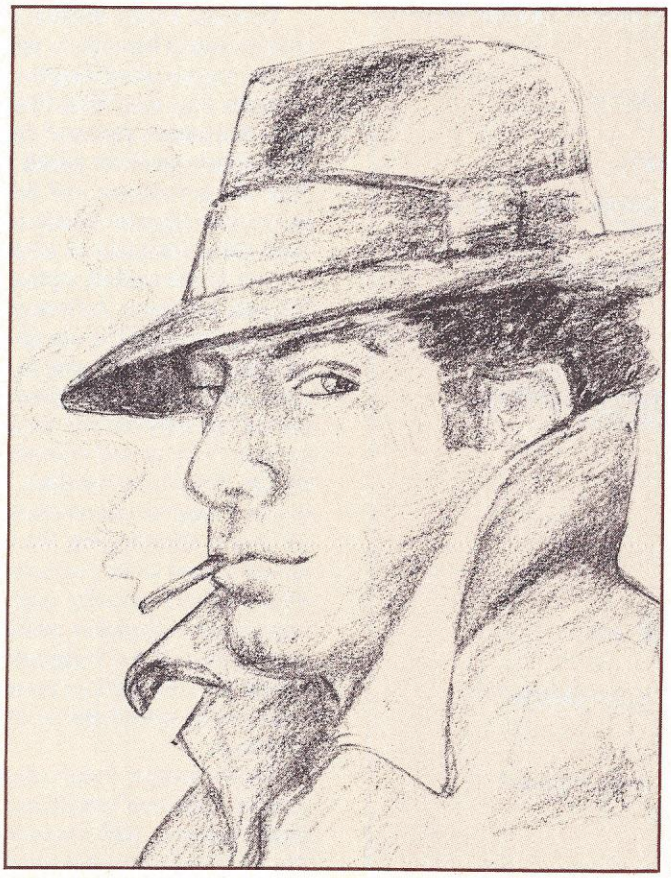


Illustration by Barbara Leonard

Security in communication and in data storage has become an important issue today. Increasing automation and networking makes it easier for clever individuals to gain access to corporate and personal records.

How can individuals and small businesses insure confidentiality of their communications? One answer lies in using codes along with microcomputers. Software can quickly and easily encode and decode messages according to complex algorithms, making your communications secure from even the most determined code breakers.

When the stakes are high enough (as in high finance, politics or war), almost nothing can stop professional code breakers. But methods being developed today – when implemented on microcomputers – make code breaking "computationally infeasible". That is, the best cryptologist (code breaker), aided by the fastest computers, would need thousands of years to decipher a message.

Every personal computer user knows a bit about coding. Before our machines can understand us, we must transform our messages using the American Standard Code for Information Interchange (ASCII). This coding algorithm or key is widely published, making ASCII useless for secrecy, but it's a code nonetheless. In fact, a one-for-one substitution, as in ASCII, is the basis for many codes.

The simplest substitution code was used by Julius Caesar to send messages back to Rome. As he wrote, he substituted the next letter in the alphabet for each letter in his message. Thus, "send help" became "tfoe ifmo". Today, a code which involves shifting letters one or more places in the alphabet is called a "Caesar substitution".

A more secure code, but also more difficult to implement, substitutes letters in a random fashion. While *B* might stand for *A* in such a code, *C* would not necessarily stand for *B*.

Writing messages directly into this code can be difficult. It's easier to begin by writing down your unencoded message – the plaintext and then using your table of substitutions to transcribe an encoded version of your message – the ciphertext. The recipient of the ciphertext would use a copy of your table to decode the message. Anyone intercepting the communication would have to duplicate the substitution table before he could read the message.

However, most amateur code breakers can decode substitution code messages, so codes of this type are still not all that secure. They're just hard enough to break to be fun; millions of people have discovered this as they solve the daily "cryptogram", a coded quotation, published by

many newspapers. With a little practice, you can solve these codes over breakfast.

Listing A

```
1 PRINT"PROGRAM TO CREATE CRYPTOGRAMS"
2 REM  WRITTEN IN MICROSOFT 6502 BASIC
3 REM  LEFT$(A$,N) RETURNS THE N LEFT MOST
            CHARACTERS
4 REM  IN STRING A$
5 REM  MID$(A$,I,N) RETURNS N CHARACTERS
            BEGINNING WITH THE I'TH
6 REM  LEN(A$) RETURNS THE NUMBER OF CHARATERS
            IN A$
7 REM  ASC(A$) CONVERTS THE CHARACTER IN A$ TO
            ITS ASCII VALUE
8 REM  CHR$(V) RETURNS THE CHARACTER WHOSE ASCII
            VALUE IS V
9 REM  C$+D$ CONCATENATES THE TWO STRINGS
10 REM
11 REM EASY CRYPTOGRAMS RETAIN WORD GROUPINGS
12 REM HARD ONES BREAK THEM INTO 5 LETTER GROUPS
14 H$="E"
15 PRINT"HARD OR EASY";
16 INPUT A$
17 IF LEFT$(A$,1)="H" THEN H$="H"
18 K=0
19 DIM D$(26)
30 PRINT:PRINT"CREATE A CODE";
40 INPUT A$
50 IF LEFT$(A$,1)="Y" THEN GOSUB 300:GOTO 135
60 IF LEFT$(A$,1)<>"N" THEN 40
70 PRINT"ENTER CODE"
71 REM TYPE "SAME" TO SAVE OLD CODE
80 INPUT A$
90 IF A$="SAME" THEN 135
100 FOR I=1 TO 26
110 D$(I)=MID$(A$,I,1)
120 NEXT I
135 PRINT:PRINT"BEGIN ENTERING PLAINTEXT"
136 REM TYPE "DONE" WHEN FINISHED
137 C$="  "
140 INPUT A$
144 IF A$="DONE" THEN  30
150 N=LEN(A$)
160 FOR I=1 TO N
170 B$=MID$(A$,I,1)
180 V=ASC(B$)-64
189 REM IS B$ A LETTER
190 IF V>0 AND V<27 THEN 200
191 IF V=-32 AND H$="H" THEN 210
192 C$=C$+B$
193 K=K+1
194 IF K=5 AND H$="H" THEN K=0: C$=C$+" "
195 GOTO 210
200 C$=C$+D$(V)
201 K=K+1
202 IF K=5 AND H$="H" THEN K=0: C$=C$+" "
210 NEXT I
220 PRINT C$
230 GOTO 136
300 REM CREATE A CODE
305 PRINT"DECODE KEY = ";
310 FOR I=1 TO 26
320 D$(I)="Z"
324 NEXT I
330 FOR I=1 TO 26
340 Q=0
350 R=INT(RND(1)*(27-I)+1)
360 FOR J=1 TO 26
370 IF D$(J)="Z"  THEN Q=Q+1
380 IF Q>=R THEN 400
390 NEXT J
400 D$(J)=CHR$(I+64)
404 PRINT CHR$(J+64);
410 NEXT I
420 PRINT:PRINT"ENCODE KEY = ";
430 FOR I=1 TO 26
440 PRINT D$(I);
450 NEXT I
454 PRINT:PRINT
460 RETURN
```

Your computer can handle encoding and decoding tasks and create a substitution table for cryptogram-type codes. (See Listing A for a program.) But actually breaking these codes is probably beyond the capabilities of microprocessors. (Still, it would be an interesting project to try.)

You can use your BASIC random number function to generate a substitution table. Plaintext entered at the keyboard is then reproduced as ciphertext. For fun, have someone else enter messages which would become cryptograms for you to solve. Or use the program to implement low security coded communications and send a friend a copy of your substitution table. Then ciphertext entered at his keyboard will reproduce as the original plaintext.

More than six billion codes can be generated by this program, so you can afford, in the interests of security, to switch codes often.

To make a code breaker's task more difficult, you can send messages through in groups of five letters rather than in the original word lengths. Clues provided by word patterns are thus lost. With this program, your personal computer becomes a personal cipher machine.

Cipher machines speed translation of messages to and from plaintext and they can employ codes far more secure than simple substitutions. Your computer can emulate these machines to achieve the same security.

One of the earliest cipher machines was invented by Thomas Jefferson. Jefferson cut eighty identical wooden disks and around the outside of each disk, he wrote the 26 letters of the alphabet in random order, a different order on each disk. He then stacked the disks and connected them by an axle through their centers loosely enough to be turned individually, but tightly enough not to slip. To use the machine, Jefferson turned the disks one at a time so the plaintext appeared in a row. Any other row then became the ciphertext. The ciphertext would appear to be a simple substitution code, but since 80 codes were actually used, the message becomes almost impossible to decipher unless the recipient had an identical machine. If the recipient had such a machine, he would turn his disks to present the ciphertext as a row and then he would search the other 25 rows to find the plaintext.

Many modern cipher machines work on the same principle as Jefferson's machine. They employ many more disks, however, and connect them in a variety of ways. They are designed so each entry advances the mechanism in some fashion, constantly changing the possible substitutions. Presented with a small amount of ciphertext from one of these machines, a code breaker wouldn't be able to deduce the plaintext. Because the pattern generated by the machine will eventually repeat, the code can be broken given a large enough sample and enough time to analyze it. To prevent this, cipher machines have adjusting dials to change the pattern daily.

The disadvantage of such machines lies in the possibility that an enemy might deduce the operation of the machine itself. Then, only the daily dial settings need to be discovered for the enemy to illicitly recover messages. For example, just before World War II, American code breakers constructed a replica of the Japanese diplomatic coding machine by using the volume of coded messages

they had already deciphered. In many instances after that, Americans intercepted and decoded diplomatic messages even before the Japanese embassy received them.

Despite their limitations, cipher machines can be effective in protecting personal and business communications. Rather than construct an apparatus of disks and gears, however, you can simulate the cipher machine with software on your personal computer. Each party to the communication only needs a copy of the same program. The software's flexibility lets you change the coding algorithm by simply altering the program. Developing such a program could become an interesting hobby application for your system.

Another technique offers even greater security. Remember how Jefferson's cipher machine increased security over simple substitutions? It used 80 substitution codes instead of only one, and repeated them only once each line. Although modern cipher machines increase that number to many thousands, the codes are still broken. Suppose then, we used a new code for every letter, and never repeated it. The code breaker is now completely foiled, because every combination of letters – every message you might send – offers an equally probable solution to the ciphertext.

You could implement this system using the BASIC program shown in Listing A without generating an entire code for each letter. As you enter the plaintext, simply shift each letter (as in a Caesar substitution) a random number of places. To recover the plaintext, the recipient employs the list of random numbers you used to create the shifts. This code, called a "one time pad", uses each pad or series of shifts only once. Each pad is, of course, just as long as the plaintext message. Furthermore, the pad must be transmitted separately by some secure means and then protected against discovery.

The pad method is effective for communications between two personal computer users. For example, suppose I have some hot new ideas for computer applications I'd like to send to the editors of *Personal Computing* in Boston but I'm afraid a rival magazine will intercept the communications.

To insure security, I encode my material using a one time pad which the editors and I previously created and recorded on a cassette tape. I play a selected block from the tape into my computer's memory and then enter the plaintext on my teletype keyboard. Although I type normally, the printed message appears to be garbage. When this "garbage" is typed into a computer containing the same pad, however, my original plaintext appears. In between, no one can decipher the message.

The program to accomplish this is surprisingly simple. (See Listing B for a sample 8080 program.) Input and output is via a serial terminal, but modifications to memory mapped I/O is possible. The code pad should be loaded in memory as indicated by the comments in the program. Remember, even if someone has a copy of the program, he cannot read your message without also having a copy of the pad you're using.

As effective as the one time pad is, it has one serious disadvantage. It requires personal contact between the correspondents to exchange the pad before they send coded messages. One technique to eliminate this requirement was devised by Whitfield Diffie and Martin Hellman, who call their method a public key cryptosystem.

A public key cryptosystem uses separate algorithms for encoding and decoding. It differs from the one time pad where the same table of shifts was used for both operations. Both encoding algorithm and the decoding algorithm must be simple operations. But for the public key system to work, it must be impossible, or at least computationally infeasible, to determine one even if you know the other.

Suppose a business person named Albert wanted to use this technique. He would need to devise a unique pair of algorithms E(9) and D(a). E(a), the encoding algorithm, would be made public – perhaps published in a directory of such algorithms. When Albert's partner Bob wanted to send Albert a secret message, he would encode it using E(a) and send the ciphertext over any public channel. Because Albert has kept the inverse algorithm secret only he can recover the plaintext. When Albert wishes to make a reply, he looks up Bob in the directory of codes and finds his public encoding algorithm, E(b). Again, the encoded message can be sent safely over public channels because only Bob knows the inverse algo-

Sample Run

```
RUN
PROGRAM TO CREATE CRYPTOGRAMS
HARD OR EASY? EASY

CREATE A CODE? YES
DECODE KEY = ONLMJPUVQCYHXEBKSFDRWZTAGI
ENCODE KEY = XOJSNRYLZEPCDBAFITQWGHUMKV

BEGIN ENTERING PLAINTEXT
? NOW IS THE TIME FOR ALL GOOD MEN
  BAU ZQ WLN WZDN RAT XCC YAAS DNB
? TO COME TO THE AIDE OF THEIR COUNTRY
  WA JADN WA WLN XZSN AR WLNZT JAGBWTK
? DONE

CREATE A CODE? NO
ENTER CODE
? ONLMJPUVQCYHXEBKSFDRWZTAGI

BEGIN ENTERING PLAINTEXT
? BAU ZQ QLN ←←←←WLN WZDN RAT XCC YAAS DNB
  NOW IS THE TIME FOR ALL GOOD MEN
? WA JADN WA WLN XZSN AR WLNZT JAGBWTK
  TO COME TO THE AIDE OF THEIR COUNTRY
?
OK
RUN
PROGRAM TO CREATE CRYPTOGRAMS
HARD OR EASY? HARD

CREATE A CODE? YES
DECODE KEY = SLYXMOEGDUBNHVJZKPFIWQRCAT
ENCODE KEY = YKXIGSHMTOQBELFRVWAZJNUDCP

BEGIN ENTERING PLAINTEXT
? NOW IS THE TIME FOR ALL GOOD MEN
  LFUTA ZMGZT EGSFW YBBHF FIEGL
? DONE

CREATE A CODE? NO
ENTER CODE
? SLYXMOEGDUBNHVJZPFIWQRCAT@
SLYXMOEGDUBNHVJZKPFIWQRCAT

BEGIN ENTERING PLAINTEXT
? LFUTA ZMGZT EGSFW YBBHF FIEGL
  NOWIS THETI MEFOR ALLGO ODMEN
?
```

rithm, D(b). The business message is secure.

If the two algorithms can be interchanged – in other words, if algorithm D could be used to encode a message which algorithm E could recover – the public key cryptosystem gains another advantage. Each message can have a signature which can't be forged. When Albert sends a message to Bob, he includes a special line uniquely identified with the message. It might contain the names of the correspondents, the date and time of mailing, the topic and perhaps the length of the message itself. This line is encoded using Albert's secret D algorithm. When Bob receives the message he decodes the signature using Albert's public E algorithm and knows that only someone who knew the corresponding D – that is, only Albert could have sent the message.

Listing B

```
0010→ONE TIME PAD PROGRAM - 8080 ASSEMBLY LANGUAGE
0011→WRITTEN UNDER IMSAI OPERATING SYSTEM
0020→RESPOND TO ? WITH
0030→  C TO ENCODE
0040→  D TO DECODE
0050       LXI B,3F3FH      FUNCTION ?
0060       CALL PRNT
0070       CALL READ
0080       MOV C,A
0090       LXI H,0030H       STARTING ADDRESS OF PAD
0095       LXI D,205FH      LIMITS OF CHARACTER SET
0100 NEXT  CALL READ        FETCH PLAINTEXT
0110       MOV A,B
0120       CPI 03H          END OF TEXT?
0130       RZ
0140       CPI 0DH          CARRIAGE RETURN?
0150       JNZ ALTR
0160       CALL PRNT        OUTPUT CR AND LF
0170       MVI B,0AH
0180       CALL PRNT
0190       JMP NEXT
0200 ALTR  INX H            INCREMENT PAD
0210       MOV A,M          GET NEXT KEY
0220       ANI 1FH          RESTRICT SHIFT TO 32 PLACES
0230       PUSH P           SAVE KEY
0240       MOV A,C
0250       CPI 34H          ENCODE OR DECODE?
0260       JZ ADER
0265→DECODER ROUTINE
0270       MOV A,B
0280       POP B            RECALL KEY
0290       MVI C,44H        RESTORE INDICATOR LOST IN POP
0300       SUB B            DECODING SHIFT
0310       MOV B,A
0320       SUB E            IS RESULT IN RANGE
0330       JP OUTP          PRINT IF YES
0340       ADD D            FIX IF NO
0350       MOV B,A
0360       JMP OUTP
0365→ENCODER ROUTINE
0370 ADER  POP P            RECALL KEY
0380       ADD B            CODING SHIFT
0390       MOV B,A
0400       SUB D            IS RESULT IN RANGE

0410       JM OUTP           PRINT IF YES
0420       ADD E             FIX IF NO
0430       MOV B,A
0440 OUTP  CALL PRNT
0450       JMP NEXT
1000 PRNT  IN 03            CHECK STATUS
1010       ANI 01H
1020       JZ PRNT
1030       MOV B,A
1040       OUT 02            CHARACTER TO TERMINAL
1050       RET
1100 READ  IN 03            CHECK STATUS
1110       ANI 02H
1120       JZ READ
1130       IN 02            CHARACTER FROM TERMINAL
1140       ANI 7FH          STRIP PARITY
1150       MOV B,A
1160       RET
```

With a widespread public key system, secure transmission of commercial information can be assured. Applications to electronic funds transfer and to storage and retrieval of personnel records would also be possible.

Where do personal computers fit into the picture? Primarily as encoding and decoding machines. As mentioned earlier, these new techniques are not easily adaptable to standard coding procedures. The algorithms take the form of mathematical functions that operate on numerical representations of the messages and can be performed in reasonable times only by computers – even modest personal computers.

One set of functions, suggested by Ronald Rivest and others at MIT, is especially promising because the same procedure can be used by every participant. Only a few parameters need to be changed to provide enough variations for everyone to have his own version.

For encoding, a message is first converted to a series of numbers. In a personal computer, the ASCII representation of the letters can be interpreted as binary digits. Each line of 80 characters can be considered as one 80-byte number in the series.

The key to the secrecy of the decoding algorithm lies in the use of very large numbers. Each number is raised to a power S, modulo a second number R. This straight forward binary arithmetic procedure consists simply of a series of shifts and adds. At each step, compare the result to the modulus R. If the result is larger, subtract the modulus from it, reducing the result to a value between zero and R-1.

(Eight-bit arithmetic is modulo 256. When the value in the accumulator reaches 2^8 it "turns the corner", beginning again at zero.)

The decoding procedure is identical, except that a different power is used. The modulus remains the same. Because the encoding and decoding powers can be used interchangeably, Rivest's functions also satisfy the requirement for signatures that can't be forged.

The key to making this system public is the size and nature of the modulus R. To insure security it should be the product of two prime numbers, each at least as large as 2^{300}. The calculations will have to be performed in 80-byte precision. For each power S used in an encoding algorithm, the code owner can use the two prime numbers to find the power T which performs the inverse algorithm. Rivest points out that finding large prime numbers is fairly easy, but finding two prime factors of a huge number is fantastically time consuming. Our correspondents Albert and Bob can safely reveal their numbers S and R, knowing it is computationally infeasible for anyone to determine the prime factors of R and thus the decoding power T.

The art of coding has come a long way since the time of Caesar. The simple code he used could be broken by any amateur today.

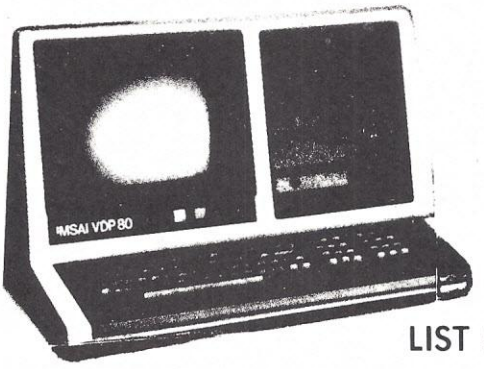
While the one-time pad provides security, it quickly becomes useless when a large volume of messages must be coded. But today, the public key cryptogram system – and personal computers – shows promise for secure communications for everyone. □

MOVING TO LARGER SPACE
SALE

APPLE II

Apple II 32K	$1,495.00
Disk II (116K) mini-floppy	495.00
Printer Interface	218.75
Expandor Printer	425.95
sugg. mfg. retail price	~~$2,634.70~~
OUR PRICE	$2,395.00

SAVE $239.30



32K RAM

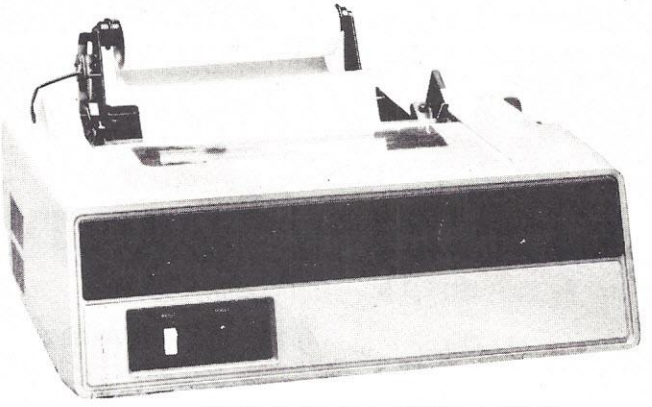
Dual Density Flopper
CRT Keyboard
8085 CHIP

The Ultimate Personal or Small Business Computer

LIST $7000.

SALE $5295.

Limited Quantity



CENTRONICS 799
Super Special $995.

ONLY $795

The fully-self-contained Commodore **PET COMPUTER SYSTEM** is still ONLY $795.00. Full 8K in RAM, 14K in ROM, full ASCII keyboard, tape drive and 9-inch CRT. Extended Basic is the language, and a machine language monitor is available on tape.
Read from Tape . . . Write to Tape. With PET's 2nd Cassette ($99.95), file maintenance is a breeze.

Now Available! **THE PET IMPACT PRINTER** . . . ONLY $595.00 . . . 60 cps . . . 8½inch paper . . . . YES, it will print the PET's GRAPHICS!

ATTENTION PET OWNERS – New products for your **PET MEMORY EXPANSION BOARD** to bring your PET upt to 32K-$595.00, RS-232 Interface 110-300 Baud (one-way)-$169.95, 110-9600 Baud (two-way)-$279.00
PET software now available:

MORTGAGE ($15.95)	ANNUAL REPORT ANALYZER ($22.95)
CHECKBOOK ($15.95)	STOCK ANALYZER ($34.95)
FINANCE ($12.95)	DATA BASE ($175.00)
OPTIONS ($24.95)	BONDS ($9.95)
STOCK SCREEN ($22.50)	SUPER STAR TREK ($19.95) SPECIAL $9.95

THE COMPUTER FACTORY

485 LEXINGTON AVENUE (46th ST), NEW YORK, NY 10017 (212) 249-1666 or (212) PET 2001 T-F 10-6 Sat. 10-4

CIRCLE 18

IN NEW YORK! FREE SEMINARS

Personal & Business Small Computer Show offers the following:

INTRODUCTORY

Introduction to Small Business Systems
Interfacing to the Real World
Introduction to BASIC
Hardware/Software/Systems Tutorial

SOFTWARE

Data Bases & File Handling
Introduction to CP/M
Introduction to PASCAL
Introduction to APL
SBS: A New Data Base Manager, Language and Operating System—a Software Breakthrough that Gives Minicomputer Power to a Micro—for Business Applications, from Computer Mart of New Jersey

SMALL BUSINESS APPLICATIONS

Text Editing/Word Processing
General Ledger/Acc'nts Receivable & Payable
Mailing Lists

RECREATIONAL APPLICATIONS

Robotics
Music
Computer Games
Art, Graphics & Animation

HOUSEHOLD APPLICATIONS

Budget Management
Investment Analysis

PROFESSIONAL APPLICATIONS

Patient & Client Billing

EDUCATIONAL APPLICATIONS

Light Pen Graphics
Simulations & Gaming

All seminars run about 50 mins. and are scheduled twice. Write for daily and hourly schedules, special hotel rates, etc. Lectures subject to change.

IN NEW YORK! AVAILABLE TUTORIALS

IEEE COMPUTER SOCIETY

Tutorials are offered by the IEEE Computer Society in conjunction with the show. Each tutorial costs as follows: members, $45; non-members, $50, and students, $35.

For further information on the technical background required, the content of the tutorials, or the IEEE, write to Dr. Daniel R. McGlynn, 71 N. Moger Ave., Mt. Kisco, N.Y. 10549

5 TUTORIALS BY SYBEX

Microcomputer Selection—Sept. 16
Intro to Microprocessors—Sept. 15
Programming Microprocessors—Sept. 16
Designing a Micro System—Sept. 16
*Interfacing Techniques—Sept. 17

*Prerequisite—Intro to Microprocessors

All Sybex tutorials run for three hours. Cost is $29.95 each up to Sept. 8. $35 each at the door. All are given in the Coliseum. Show ticket is needed for entry. Please use coupon for advance ticket.

(For registration and more details write to Sybex, 2020 Milvia, Berkeley, CA 94704) TEL: (415) 848-8233

GENERAL ATTENDANCE INFORMATION

The show will occupy the fourth floor of the New York Coliseum located at 59th St. & Columbus Circle. There is a parking garage in the building and the site is served by many subway lines and buses.

Visitors from out of town may obtain special hotel rates from our headquarters hotel, the Barbizon Plaza, just two blocks from the show. Singles $44, doubles $52. Identify yourself as attending the show and request those rates.

Admission to the show at the door is $5.00 per day, per person. Discount tickets for groups of ten or more available in advance only by mail at $4 per person per day. Enclose check or money order for all ticket orders payable to Personal & Business Small Computer Show. For further information call (212) 753-4920.

PERSONAL & BUSINESS SMALL COMPUTER SHOW

78 EAST 56TH STREET
NEW YORK, N.Y. 10022

New York Coliseum
Sept. 15–17, 1978

SHOW HOURS

Sept. 15—noon to 8 p.m.
Sept. 16—10 a.m. to 8 p.m.
Sept. 17—noon to 7 p.m.

2nd Annual

PERSONAL & BUSINESS SMALL COMPUTER SHOW

NEW YORK COLISEUM
September 15-17, 1978

EXCITING

That's the word that sums up the dynamic small computer show in New York that was established last year at the Coliseum. The lecture areas were filled by interested people, the exhibits were great, the attendance was the highest of any small computer show in the country.

The big point is that everyone who came went away happy. Accountants, hobbyists, lawyers, doctors, brokers, retailers, business people, programmers, research scientists and just plain homeowners.

The second big point is that we're doing it again. This time, bigger and better. Top seminars, top exhibits. Small computers from $500 up, software and kits.

Save time. Get your tickets in advance by mail.

NAME _____
(please print)

ADDRESS _____

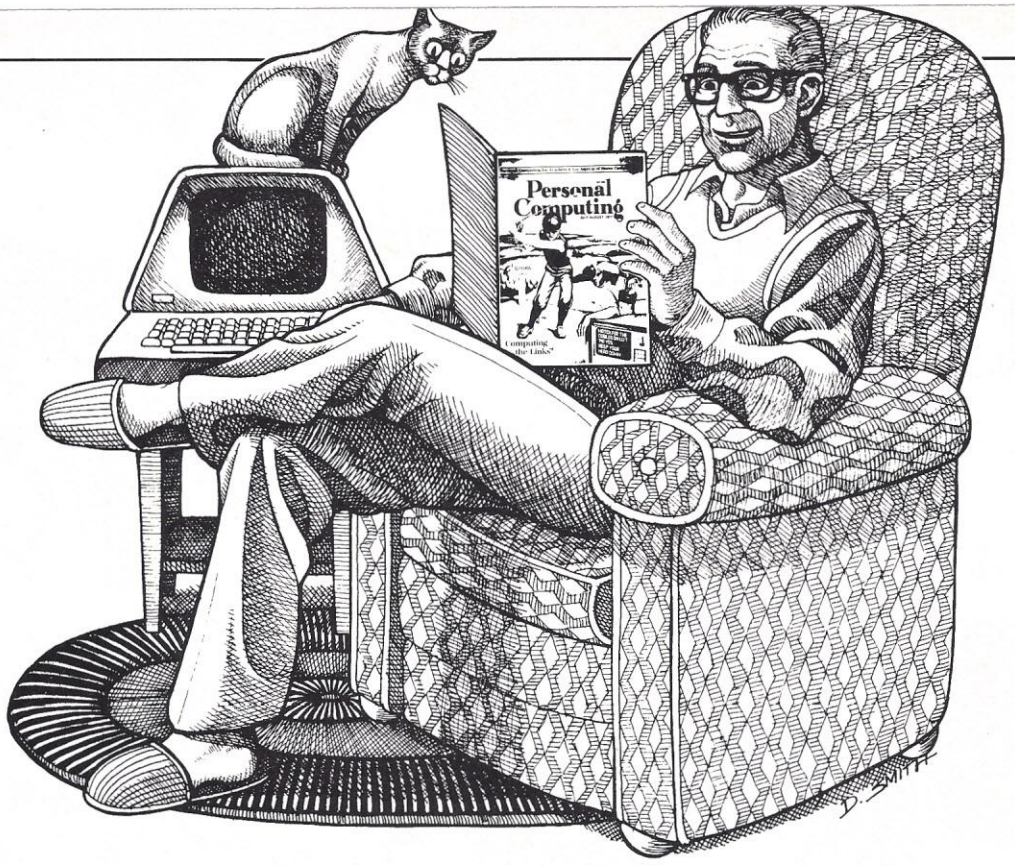
City _____ State _____ Zip _____

Send me tickets in advance for Personal & Business Small Computer Show, Sept. 15-17, 1978, in the New York Coliseum.

(Check One) One Day ($5)__ Two Days ($9)__ Three Days ($13)__

Send check or money order (U.S dollars) payable to Personal & Business Small Computer Show to 78 E. 56th St., New York, N.Y. 10022.

PERSONAL & BUSINESS SMALL COMPUTER SHOW
78 East 56th Street, New York, N.Y. 10022



Leading the way to computer mastery . . .

Maybe you own a computer. Maybe you're thinking about buying one. Or perhaps you're just curious about the computer world. Let *Personal Computing* show you how to use microcomputers for fun and profit.

A year's subscription brings you 12 monthly issues filled with informative, easy-to-read articles on all aspects of personal computing. Games to play, money-making ideas and applications and programs for home, business and educational use are just a few of the topics covered in *Personal Computing.*

We provide instructive articles on computer basics as well as more advanced articles.

So learn to master your computer the easy way; curl up in your favorite chair with *Personal Computing.* Use the attached coupon to order your subscription today.

Personal Computing Magazine

I've been bitten by the bug. Enter my subscription to Personal Computing today!

NAME_____

ADDRESS_____

CITY_____ STATE_____ ZIP_____

USA

□ 1 year (12 issues) $14
□ 2 years (24 issues) $26
□ 3 years (36 issues) $38

□ Bill me
□ Check enclosed
(you'll receive one extra issue for each year!)

Charge my:
□ Master Charge
□ Bank Americard

Account # _____
Card expiration date _____

ADDITIONAL POSTAGE (per year)

CANADA & MEXICO	$4.00 surface	$8.00 air
OTHER FOREIGN	$8.00 surface	$36.00 air

(Please remit in US funds – Thank you)

BACK ISSUES $3.00/COPY
(Payment must accompany order.)

Mail to: PERSONAL COMPUTING, 1050 Commonwealth Ave., Boston, MA 02215

Toronto Tournament Concludes

. . . . The Toronto Chess Tournament came to a close at the end of the fourth round. Before the tournament started each team had been seeded according to past performances. The following table compares the finish in the tournament to pre-game seedings. The big upset, of course was Russia's *Kaissa.* It came into the tournament seeded #1 because it had won last year's tournament. This time, however, it finished second, tied with *Duchess* which had been seeded 9th. Winner of the tournament was *Chute 4.6* which came into the fray with a seeding of 2. The complete final standings and previous seedings are shown in chart on the left.

Games at Toronto not listed previously will be upcoming in future issues with appropriate annotations.

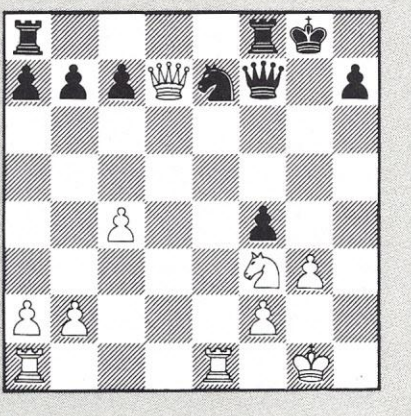
At the end of the tournament, an exhibition game was arranged between *Chess 4.6* (new champion) and *Kaissa* (previous champion) because they had not met each other in regular tournament. The game between the two teams was purely an unofficial demonstration in Artificial Intelligence and, according to the spectators, proved nothing. It is difficult, however, to match two computers in a contest without arousing the gaming instinct in the programmers. The challenge match follows:

Seedings and Final Standings

Program	Country	Finish	Seeded	Final Score Points
Chess 4.6	USA	1	2	4
Duchess	USA	2 (Tie)	9	3
Kaissa	Russia	2 (Tie)	1	3
Belle	USA	4 (Tie)	13	2½
Chaos	USA	4 (Tie)	4	2½
Master	England	6 (Tie)	3	2
Black Knight	USA	6 (Tie)	5	2
Dark Horse	Sweden	6 (Tie)	16	2
Elsa	Germany	6 (Tie)	15	2
Wita	Canada	6 (Tie)	14	2
Blitz V	USA	11 (Tie)	7	1½
Ostrich	Canada	11 (Tie)	6	1½
B.C.P.	England	11 (Tie)	10	1½
Chute 1.2	Canada	11 (Tie)	11	1½
BS '66 '76	Netherlands	15	12	1
Teli	Switzerland	16	8	0

1. P-K4	N-QB3	24. K-N2	K-N2
2. N-KB3	P-K3	25. N-N5	R-Q7
3. P-Q4	P-Q4	26. R-QN1	R-B7
4. B-Q3	PxP	27. P-N3	N-K4
5. BxP	B-Q2	28. R-KR1	RxP/R
6. 0-0	N-B3	29. R-R4	N-Q6
7. R-K1	NxB	30. N-R3	R-N7
8. RxN	B-K2	31. P-KN5	K-N1
9. P-B4	P-B4	32. NxP	RxPch
10. R-K1	0-0	33. K-N3	RxN
11. N-B3	P-B5	34. RxR	NxR
12. Q-Q3	Q-K1	35. KxN	K-B2
13. P-KN3	PxP	36. P-N4	K-K3
14. P/RxP	Q-B2	37. K-K4	P-QR3
15. B-B4	P-KN4	38. K-B4	K-Q3
16. P-Q5	PxP	39. K-K4	P-B4
17. NxPQ5	PxB	40. PxPch	KxP
18. NxBch (see fig)	NxN	41. K-Q3	P-QR4
19. QxB	N-N3	42. K-B3	P-R5
20. QxQ	RxQ	43. K-Q3	K-N5
21. P-KN4	R-Q2	44. K-B2	KxP
22. R/R-Q1	R/1-Q1		
23. RxR	RxR		

White - Kaissa Black - Chess 4.6



Position after *Kaissa*'s 18th move. The agressive *Kaissa* has opened his board for freedom of movement. It is apparent at this point that *Chess 4.6* prefers to play a defensive game waiting patiently for restless *Kaissa* to execute a weak maneuver.

Kaissa resigns.

Chess seminar

. . . . Having won the world's computer chess championship from Russia's *Kaissa* (on points) *Chess 4.6* will be at Personal Computing's 2nd Annual Midwest Exposition in Chicago, Oct 5-8 to participate with its authors David Slate and Larry Atkin in a seminar on computer-chess and demonstrate the capabilities of its program. Carnegie-Mellon, a few years ago, ran the following report in one of its journals regarding the proposed match-bet between Chess 4.6 and David Levy:

"*CHESS 4.6*, written by David Slate and Larry Atkin of Northwestern Uni-

versity, has recently played 16 games in three chess tournaments against human players; it won two of those tournaments. This performance entitles it to a United States Chess Federation (USCF) rating of 2070, which falls in the "Expert" category. By comparison, Levy is rated 2375, and Bobby Fischer is rated 2800. *CHESS 4.6* has beaten players with ratings as high as 2100, and Levy has lost to players with ratings lower than 2100, so it is not a foregone conclusion that Levy will win. The rules of the bet are that if Levy loses or ties, he is entitled to a rematch, but if Levy wins, the computer is not entitled to a rematch this session. Many thousands of dollars are involved in the wager, which Levy stands to win if he does not lose to a computer before August of 1978. Dr. Hans Berliner, of Carnegie-Mellon's Computer Science Department, himself an International Master and former world correspondence chess champion, believes that the odds of the computer winning are about one in ten.

"Computer scientists study chess and write chess programs as part of the field known as Artificial Intelligence (AI). AI researchers study chess for the same reason that geneticists study fruit flies: it is a comparatively simple and well-understood problem that can serve as a test-bed for new techniques in computer problem-solving. Chess has been the subject of intensive human study for hundreds of years, and more is known about it than many other complicated fields of endeavor. Carnegie-Mellon is one of the world's major AI research centers, and many of the techniques used by *CHESS 4.6* were pioneered at CMU in the late 1950s.

"The techniques used by *CHESS 4.6* are not so much innovative as they are the result of combining careful refinement of technique with a tremendously fast computer. The essential methods used by *CHESS 4.6* have been known for years: a computerized representation of knowledge about chess, and various techniques for using that knowledge to play the game. Analogously, the aerodynamic principles used in designing the Concorde supersonic transport have been known since the late 1940's, but only recently, by combining careful engineering with a super-powerful jet engine, were they used to create the first supersonic passenger plane.

"In analysis of the playing record of *CHESS 4.6*, Dr. Berliner notes that its tactics are superb, of Grandmaster quality, but that its strategy and long-term planning ability are only mediocre. As faster computers are built and better strategies are developed, the program can only get better."

Echoes of the past

. . . . In connection with the current struggle of the computer trying to surpass the human in chess it is appropriate to recall extracts of an article by Benjamin Mittman (of Northwestern University and director of its Vogelback Computing Center) which appeared in Datamation in 1973 and is reprinted here with permission from Prof. Mittman.

"Can a computer beat Bobby Fischer? This question dominated a panel discussion which I chaired at ACM's 1972 annual conference. Members of the panel included David Levy, an international chess master; Monroe Newborn and George Arnold of Columbia; Edward Kozdrowicki of Univ. of Calif. at Davis; David Slate of Northwestern; Franklin Ceruti, of the US Air Force; Albert Zobrist of USC; James Gillogly of Carnegie-Mellon and Bruce Leverett of Harvard. In addition to the panel members, we were fortunate to have an audience which contained people who were interested in the subject and who had worked on these and related topics. Among them was Dr. Arthur Samuel of Stanford and formerly of IBM who is the developer of the world famous, master-level checker playing program. The issues discussed in the panel included strategy vs. tactics, learning and advice taking, creativity, and the value of game playing research.

"The better programs are usually tactically sound. They do not blunder in obvious ways; they do not fall prey to simple 'cheap shots'; they can even produce an occasional 'brilliant' move. They accomplish this by applying effective chess knowledge while evaluating a position and following the consequences of these evaluations to five, six, seven or eight ply in their tree search. But what about chess strategy?

"Can the program 'decide' on mounting a queen-side attack with a pawn storm after having first prepared the ground with knight migration to the queen side and by opening an appropriate file for the eventual doubling of rooks? Can the program divert attention from a strong point which it eventually 'plans' to occupy by creating pressure first on some other portion of the board? The current crop of programs cannot do these things.

"When the program moves a knight to B3, it does so because the book opening dictated that move or because the evaluator highly valued a move that attacks the central squares. To retreat this knight for strategic reasons requires a level of analysis that transcends pure tactics. It would require that a strategic analyzer be invoked when the tactical analyzer finds no sufficiently strong moves to 'suggest.' This should be done. But, to date, it hasn't been done.

"The panel members agreed that strategic play is an absolute necessity for master-level chess. Little progress has been made since all of the programmers have been busy producing programs that play reasonably effective tactical chess. But strategy is not all that is required to beat Bobby Fischer, or any other master or grandmaster, for that matter.

"To play master-level chess, a program must be 'taught' or must 'learn' tactics and strategy, openings, end game play, fundamental positional ideas, king safety, pawn structure, zug-zwang, etc. During the panel Dr. Samuel stated that he would have liked to have had a checkers master *teach* his program if he had an adequate means of allowing the master to *communicate* with the

program. Unfortunately, his program and most of today's chess programs must be communicated with through some programming language.

"Dr. Zobrist and others on the panel felt that the pattern recognition approach being taken at USC offers such a possibility. Charles Kalme of the USC team is a master-level chess player, but not a programmer. He was able to utilize the pattern language available in the USC program to input the 45 patterns which were used at ACM '72. Unfortunately, USC only won one game in Boston. However, the consensus of the panel was that the play of the USC program would improve considerably with a larger repertoire of patterns. Rather than 'learning,' Dr. Zobrist referred to this mode of pattern processing as 'advice taking.' David Slate of Northwestern (with an 'expert' USCF rating) felt, also, that substantial improvements to Northwestern's program must await a more effective procedure for inputting chess heuristics – for advice taking.

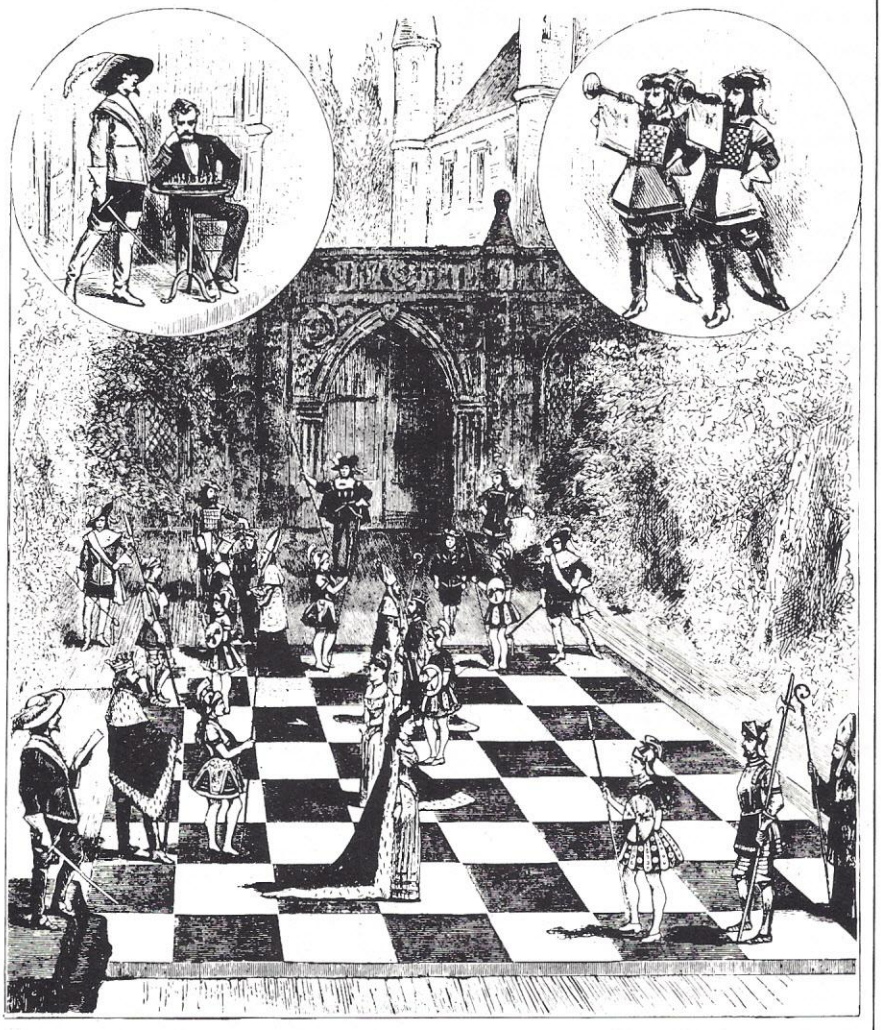
"David Levy was skeptical of the claims that a chess playing program could beat him in a match by 1978 (he has put £1000 behind this skepticism in bets with four computer scientists). He conceded that strategic play could be implemented. He agreed that improved advice-taking procedures and pattern processing could improve play beyond the current Class C level. However, Mr. Levy doubted that master-level chess could be achieved. The issue of creativity was brought up – a topic of current interest, since Bobby Fischer and Boris Spassky were demonstrating creativity across the chess board in Reykjavik.

"One might define creativity at the chess board as a process in which surprising, unorthodox, unexpected, and new moves are made which prove to be strong after further play and/or further analysis. The problem with today's computers and computer programs is that they are unable to 'see' the potentially interesting patterns without deep analysis. The capacity of the human chess master to screen out uninteresting lines without careful analysis, i.e., the ability for *gesttalt* pattern recognition is a difficult barrier for computers to overcome.

"David Levy contended that no program could display creativity, when looked at in this way. However, David Slate said that he believed that computer algorithms will be developed to detect the kinds of features and combinations of features of board positions, whose recognition will enable the computer to select creative, deep and ingenious moves with an efficiency approaching that of a chess master. The question of creativity in chess playing is thus one large unknown in discussing the question of whether a computer can beat Bobby Fischer.

"One final subject which interested the panel, but which has little to do with the subject of this article, concerned the value of game playing computer research. This was not a question that really concerned the authors of the programs, since all of them would tell you that they have put all those man-years into writing chess programs because it is fun. Dr. Samuel urged them to continue their efforts because he feels that solving chess playing or checker playing problems is one step toward solving even more difficult problems. He personally has moved on to research in speech recognition. Speech recognition must deal with far more complex patterns, far more complicated decision trees, and far more varied cognitive environments than any game. But progress in the development of tree searching algorithms, pattern recognition, decision making, and other techniques in artificial intelligence research is sure to further progress in speech re-

A Game of Living Chess



A game of living chess at the Academy of Music, New York, in 1879.

cognition. "The annual ACM tournaments have generated enormous interest in the public press – helped, of course, by the Fischer-Spassky world championship. The authors of the programs are intent on improving the level of play of their programs; however, some of them feel that the end of the road has been reached using the Shannon structure. Thus future efforts will be directed toward the areas discussed in the panel. As a result, it appears that a new era in computer chess playing research has begun. David Levy will probably win his bets, but there are many computer scientists who would answer the question 'Can a Computer Beat Bobby Fischer?' with 'Maybe.' However, not many of them are willing to answer the next logical question: 'When?'"

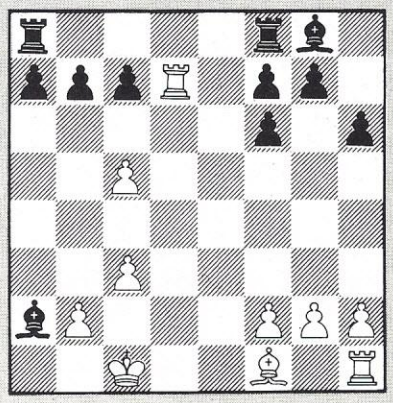
A Difference of Opinion

. . . . A letter from H. Lyman, Boylston Chess Club, Boston, points out a differing opinion to Dr. Swets' explanation of the *Duchess-Kaissa* game at the Toronto tournament (see Personal Computing April 1978). "I note one error in Dr. Swets' description. Dr. Swets had intended to state: '*Kaissa* than plays . . . . 34 R-K1. A very weak move at first sight . . . etc.' This particular position has become famous throughout the chess world as a result of its appearance in the British Chess Magazine with the following paraphrased account: 'Upon making . . . . 34. R-K1, Mikhail Botwinnik, the Soviet ex-world champion commented publicly that if one of the best computer chess programs in the world *(Kaissa)* could blunder a rook away by . . . 34 R-K1, then computer programming in chess had a long way to go. David Levy who was MC'ing the event generally concurred with Botwinnik's commentary. Subsequently *Kaissa* was 'asked' why . . . 34. R-K1 was made. Its explanation was that given on page 106 (Personal Computing), 'mate in five moves.' The computer was smarter than Botwinnik!!"

At Seattle's Tournament

. . . . The Seattle Tournament wound up in a tie for first place between *Chess 4.6* (flushed from a recent championship victory at Toronto) and *Chaos* of the University of Michigan. At the conclusion of the regularly scheduled Swiss-style tournament David Levy, international chess master, played a four-hour exhibition match against all 12 computer participants. David won 10 of the matches either by mate or resignation, managed a draw against *Chaos* and lost to *Chess 4.6.* The game that Levy lost follows:

White: Levy	Black: Chess 4.6
1. P-Q4	N-KB3
2. N-QB3	P-Q4
3. B-N5	P-KR3
4. BxN	KPxB
5. P-K4	PxP
6. NxP	N-B3
7. P-QB3	Q-K2
8. Q-K2	B-K3
9. N-B5	B-Q4
10. O-O-O	QxQ
11. NxQ	BxN
12. PxB	BxRP
13. N-Q4	NxN
14. RxN	O-O
15. R-Q7(see Fig)	QR-B1
16. P-QB4	B-N6
17. K-Q2	KR-K1
18. B-Q3	B-R5
19. R-Q4	B-B3
20. P-B3	R-K4
21. P-QN4	R-N4
22. B-B1	P-QR4
23. P-R4	R-K4
24. P-N5	B-K1
25. R-Q5	R-K2
26. R-Q4	P-B3
27. P-N6	R-K4
28. B-Q3	RxP
29. R-K1	R-K4
30. R/4-K4	RxR
31. BxR	P-R5
32. K-B3	B-Q2
33. P-B5	P-R6
34. P-N4	B-K3
35. B-B5	BxB
36. PxB	R-Q1
37. R-QR1	R-Q4
38. K-N4	RxP/KB4
39. P-B4	RxPch
40. KxP	R-B5
41. R-Q1	RxBP
42. R-Q8ch	K-R2
43. R-Q7	R-R4ch
44. K-N4	R-QN4ch
45. K-B4	RxP
46. RxBP	K-N3
47. R-Q7	P-KB4
48. Resign	



In the opening moves, *Chess 4.6,* refusing to be intimidated, matched Levy's play with equal aggresiveness. In the first 14 moves, 8 major pieces (in addition to 3 pawns) had made hasty exits showing that the two adversaries had, in the manner of a boxing match, "come out slugging." After this initial flurry of furious activity both champion players settled down to a more cautious game.

San Jose Micro Tournament

. . . . Larry Wagner, tournament director of the first microcomputer chess tournament, held at San Jose, CA in March 1978, sends along the following report: "For the 2½ days of the Faire a little less than a dozen microprocessors (and their memories, I/O, power supplies, etc.) along with their programmers competed in a computer versus computer chess tournament. This was the first computer chess tournament *without* terminals and telephone hook-ups to remote machines whose values could be measured in millions of dollars. The most expensive computer entered was around $6000 while the lowest cost entry was a 'homebrew' metal box containing $85 worth of parts. When it was all over *Sargon,* a program for a Z-80 developed by a husband and wife programming team, finished in first place with a convincing 5 wins out of 5 games played. After the field of entries was established, the class structure was defined to be:

CLASS A	MICROCOMPUTERS WITH 8K OR GREATER MEMORY
CLASS B	MICROCOMPUTERS WITH LESS THAN 8K MEMORY
CLASS C	PROGRAMS RUNNING IN BASIC

"The prize certificates were awarded by the class. However, all the entries were played against each other. As the tournament progressed, we had to make adjustments in the rules and scheduling. During the first day, two of the programs running in BASIC could not keep up with the time control of 50 moves in 2 hours. It was decided that they would play each other in a single 9 hour match on the second day while the other competitors were playing in two 4 hours matches. Several matches had to be restarted due to loss of power when someone pulled out a plug by mistake, or the realization that there was a notation misunderstanding, or for undetected move entry errors.

"We noticed that the machines had a tendancy to make repeated moves or perpetual checks even when they were significantly ahead in the material. In order to prevent these games from resulting in a draw, we allowed the programmers to adjust the machines to increase or decrease the look-ahead level or response time to try and get out of these 'lockup' situations. At the conclusion of the computer versus computer tournament, Alan Benson, Senior Regional Vice President of the United States Chess Federation and ICCF Postal Master, played a simultaneous exhibition against all the computers (and a few humans too). Following is one of the game scores (*Sargon* vs. Steve Stuart) with Alan's chess commentary.'

White: Sargon	Black: Steve Stuart
1. P-Q4	P-Q4
2. N-QB3	N-QB3
3. B-B4	B-B4
4. N-B3	N-B4
5. P-QR4(a)	P-K5(b)
6. PxP	N-KN4
7. P-R3	KNxPK5
8. BxN	NxB
9. NxN	P-Q5
10. P-KN4?	B-K3?(c)
11. N-N5	B-N5ch
12. P-QB3	B-QB4
13. PxP	B-N5ch
14. N-B3	Q-O
15. B-N2	P-QB3
16. Q-Q3	BxNch
17. QxB	Q-Q3(d)
18. O-O	P-B3
19. P-K3	Q-QB2
21. KR-Q1	QR-Q1
22. P-B4	K-R1
23. K-B2(e)	P-QR1
24. P-B5	B-N1
25. P-K4	Q-B5ch
26. K-K2(f)	KR-K1
27. N-R4	R-K2
28. Q-B5	R-Q3(g)
29. N-B4	QR-Q2
30. N-N6	Q-N6
31. R-KN1	R-Q3(h)
32. N-B8(i)	R-Q1(j)
33. NxR	B-N6
34. R-R3	P-KN5
35. Q-N6(k)	R-Q2
36. RxB	Q-R7
37. N-N6ch(1)	PxN
38. K-B2	Q-B5ch
39. K-K2	Q-R7
40. K-B2	Q-B5ch
41. K-K2	Q-R7
42. P-Q5(m)	KNPxP
43. PxQBP	R-K2
44. Q-Q1ch	R-K1
45. QxRch	K-N2
46. RxPch	K-R3
47. Q-R5 mate	

Alan Benson's Annotations

(a) In *Sargon's* programming these rook pawn moves appeared often. Normal would be 5. P-K3.
(b) A miscalculation.
(c) Missing a golden opportunity with 10 . . . PxN! 11. PxB (Best is 11. PxP immediately giving back a piece) 11 . . . . PxP and now white's best is 12. N-Q3 PxR=Q 13. QxQ since 12. R-N1 B-N5ch and mates next move, or 12. QxQ1ch RxQ13. R-Q1 B-N5ch 14. R-Q2 P-N8=Q mate.
(d) This move eventually loses a tempo. Practically speaking black should play 17 . . . . Q-B2 followed by developing the rooks to the center files.
(e) Strange for the white king to be taking a casual walk into the center.
(f) 26.K-N1 would be much safer.
(g) Here 28 . . . . QR-K1 followed possibly by 29 . . . . B-Q4 putting pressure on white's KP would have given the best practical chances.
(h) 31 . . . . R-QB2 was better.
(k) Very nice – attacks both the rook and bishop.
(l) Also possible was 37 K-B2 Q-B5ch 28. R-KB3 Q-Q7ch 39. K-N3 RxP 40. Q-B7 R-Q6 41. Q-B8ch(If 41 RxR?? Q-B5 mate) 41 . . . . R-Q1 42. Q-K6 winning easily.
(m) *Sargon* was allowed a three move look ahead with this move to avoid the repetition of moves. It plays the final part of this game very well.

Micro Chess Champs

. . . . Kathe and Don Spracklen have published the literature on their *Sargon* program. This program runs on a Motorola 6800 microcomputer and won first place at the San Jose Microcomputer Chess Tournament. The material consists of 53 pages of assembly language listing of the program and is accompanied by 100% commentary on the performance. Doug Penrod has seen an advance copy of this publication and is preparing a book review on it. The program is available from The Spracklens, 10832 Macouba Place, San Diego, CA 92124. Price of the program is $15 but well worth it, says Doug, especially to anyone interested in transforming his own microcomputer into a chess-playing machine. How thoroughly the Spracklens have documented the Sargon program, says Doug, can be seen from the table of contents, reproduced here.

Table of Contents

Scotland Report

. . . . Ronald W. MacRae, of Crawley, Sussex, England has sent along this note regarding a conference/exhibition on "Advances in Computer Chess" which was held in Edinburgh University, April 10th and 11th. "The conference was sponsored by IBM (UK), ICL, Computer Weekly and The Times and Sunday Times newspapers. Five of the papers presented concerned 'knowledge refinement,' that is, reduction of detailed knowledge to correct and economical algorithms. Two of these five dealt with routines to avoid database look-up in the end game of king and pawn against king. Two others dealt with pattern representation in general and games and reduction in specific of the king and rook against king and knight database (the one which revealed that in certain positions it was possible for black against optimal play to postpone checkmate beyond the 50 moves required by some tournament rules) to an advanced table. The fifth paper and last of the conference expanded the concept of knowledge refinement from chess to other fields and was illustrated with examples drawn from medical, chemical, industrial and botanical computing. Two papers were presented by P. Kent (Rutherford Lab) and J. Birmingham (UK Atomic Energy Research, Harwell) on *Master,* one of the top chess programs in the world. Main points were the development, history, top level design and detection of checkmate at shallow ply. (*Master* could detect mate at above 3 ply but not always when closer.) In the evening, M.R.B. Clarke won against *Master* in a match played under H. Golombek's supervision.

Part IV of Valenti Thesis on Chess

This section gives a detailed account of the design and implementation of CHUTE I (CHess, University of Toronto, Engineering). The main purpose is to supply an easily modifiable framework of a chess playing program that allows different ideas or strategies to be inserted and tested. In the process of implementing this framework, much of it has been filled out, to the point that it also plays a reasonable game of chess, but the facility remains to replace or add to these strategies.

Programming conventions

A number of programming conventions were adopted that provided for easy understanding and readability of the program listing. The main feature of these conventions is that they are consistently followed throughout the entire program. The basic conventions, relating to the structure, are:

1. The program structure diagram showing the nesting of all procedures appears at the start of the program and is updated whenever procedures are added or deleted.
2. Procedure main bodies and declarations are not indented.
3. Indenting starts in column 2 and uses every 4th column (2,6,10,14,. . .).
4. Line continuations are indented 4 columns.
5. 5 blank lines are inserted before and after each procedure.
6. Variables I,J,K,L,M,N are declared FIXED at the start of all non-trivial procedures requiring temporary data items. This is followed rigidly to avoid complications where an enclosed procedure uses the temporary variables declared in the outer procedure.
7. All enclosed "DO WHILE"s and iterative "DO" groups have their enclosed statements indented.
8. "IF <expression> THEN . . . ELSE . . ." is split up thusly:

IF <expression>
THEN
  <statement>
ELSE
  <statement>

9. If there is no "ELSE" clause in an "IF" statement, the keyword "IF" is replaced by "WHEN".
10. Simple "DO" groups within "IF" statements have the "DO;" and "END;" each starting in column 75 and the body of the group is indented four columns from the keyword "IF".
11. A few variables are declared to be literally '/*' or ' ' in order to make some debugging statements compiled as code or listed as comments. These declarations appear at the start of the program listing and the areas in the code are marked with comments as follows:

START OF OPTIONAL OUTPUT AREA SSSSSSSSSSSSSS
(optional code)
END OF OPTIONAL OUTPUT AREA EEEEEEEEEEEEEEEE

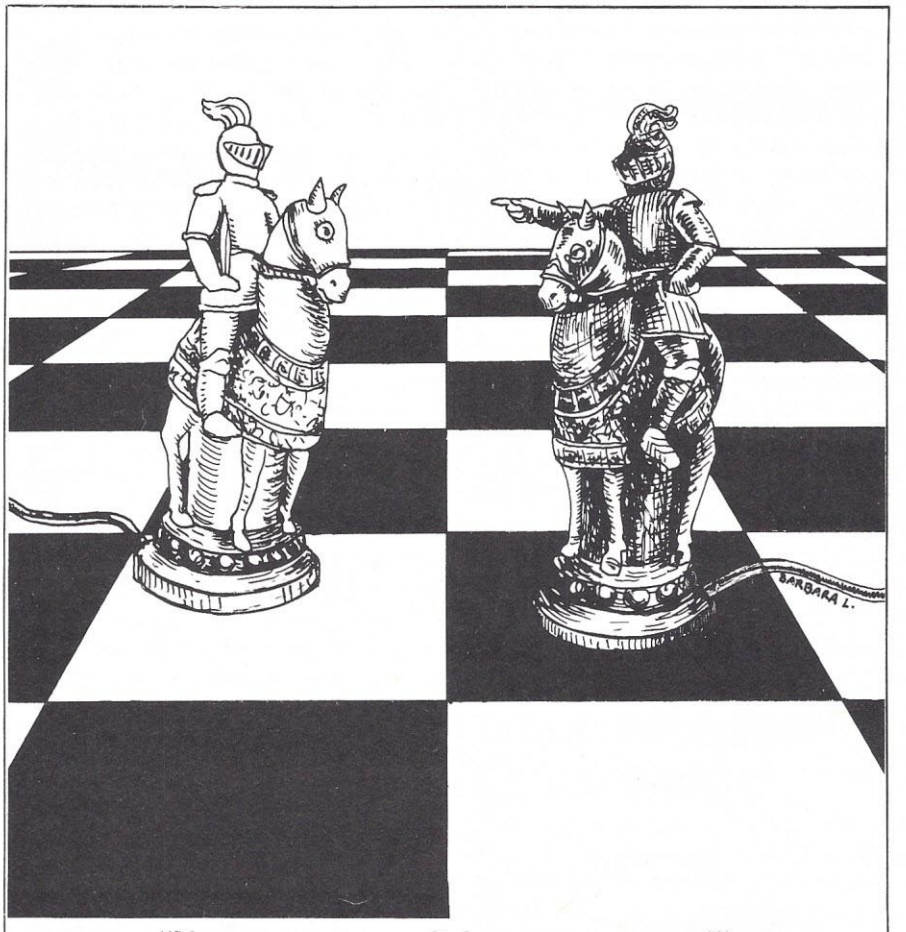
12. All variables used in a procedure are declared at the beginning. Enclosed procedures follow immediately, before any executable code. Comments for the procedure usually appear after these.

One feature in BPL is that no variables may be referenced before they are declared. For this reason, enclosed procedures must precede the code that uses them. Therefore, the main control procedure appears at the end of the program listing.

Information retrieval

This program is basically made up of decision-making processes in the plausible move selection and heuristics routines, namely, the chess-specific areas. These decisions are based on features of the board position that are sometimes easy to find (the colour of a piece) or they are more difficult (what pieces are pinned?) The problem is fetching this data as quickly as possible, without duplicating effort if this information is required in more than one place.

Some information can be retrieved



"My programmer can lick your programmer!"

very quickly and there is no reason to worry about duplicating efforts. For instance, the colour of a piece is given by referencing a vector with the piece number as an argument (pieces are assigned numbers from left to right, top to bottom as they appear on a board, except that the kings are numbered last). But other data requires more effort to obtain, and should be retrieved or computed once and then placed in an easily accessible format for future reference. An example of this is computing what pieces are pinning others. A pin is when one piece is attacking another piece indirectly, with a piece intervening. This is computed once and the information is stored in a 64-byte vector and can be accessed with a square number, to obtain information on pieces pinned on or through that square.

A decision must therefore be made as to whether it is worth re-evaluating data each time it is required, or to take the time to reformat the data for easier access later on. Basically, the information is most easily retrieved from tables (vectors or arrays) and these tables can be (usually) easily made from the information in the chess position data structure.

An example of information reformatted for future reference is the location table, LT(64) which contains 64 16-bit entries, one for each square on the board. Each bit position within an element of the vector is treated as a flag for a certain situation, such as good locations for castle pawns, squares around the enemy and friendly kings, passed pawns and the squares in front of them. Once in this form, the information is easily found by testing the appropriate flag.

Reorganizing this data usually makes the program more readable, because when data is precollected, a relevant name can be given to it. For instance, to check that square "x" is a square in front of an enemy passed pawn, the following is used (using the language BPL):

"IF" tests the low order bit of the expression
"SHR" shifts the first argument right the number of places specified by the second argument
"ENEMY-PP" is a name given to some constant
"BOARD-VECTOR (64)" is a 64 byte vector giving the piece codes on each square
"BBPM" = black passed pawn (moved)
"BPPNM" = black passed pawn (hasn't moved)
"WPPM" = white passed pawn (moved)
"WPPNM" = white passed pawn (hasn't moved)

```
IF SHR (LT(x), ENEMY-PP)
THEN
. . . etc.
```

instead of:

```
IF TURN = WHITE
THEN
DO;
IF (BOARD-VECTOR (x-8) = BPPM)
   (BOARD-VECTOR (x-8) = BPPNM)
THEN
-etc-;
END;
ELSE /* TURN = BLACK */
IF (BOARD-VECTOR (x+8) = WPPM)
   (BOARD-VECTOR (x+8) = WPPNM)
THEN
. . . etc.
```

(The board is numbered left to right, top to bottom with white at the bottom. This example doesn't worry about checking near the edges of the board. In actual fact a special 12 by 12 board is used to avoid referencing off the board.)

The first example above shows a more concise, but still easily understandable, and more efficient representation of the code to perform this specific task. This example may be a little extreme, but it illustrates the point that where and how information is obtained is very important to the performance and understandability of the program. (Next month we consider "Chess Position Data Structure.")

Mississippi Chess

. . . . Warren B. Porter, of Jackson MS, sends along this extract from the June issue of the Mississippi Chessletter which features a game between *Blitz V,* University of Southern Mississippi and *Duchess,* Duke University's program.

Blitz V, says Warren, was developed by Robert Hyatt of USM in Hattiesburg. The program runs off a large Honeywell computer and the source program is about 10,000 cards long in Fortran. Future plans for *Blitz V* are to convert it to run on the IBM 370-155, when funds become available. During the computer tournament in Toronto, *Blitz V* finished in a tie for 6th place while *Duchess* (see below) tied for second with *Kaissa.* The tournament held in April of 1977, Warren writes:

"One of the highlights of the Association of Computer Machinery conference held in Biloxi during April and hosted by USM was a tournament of computers. Participating in the tournament were *Duchess,* a program from Duke University, the Greenblatt program from Ole Miss, and *Blitz V* from USM. When Blitz wasn't playing other programs it played against people who wanted to take it on during the conference.

"*Blitz V* was written by Robert M. Hyatt, Chief of Systems Programming at USM. Dr. Hyatt's own introduction to the program, taken from the Proceedings of the 15th Annual Southeast Regional Conference is as follows: *Blitz V* is the latest of a two year old series of chess programs. The strength of the program has increased gradually as the program has been developed; it currently wins over 90% of the games played with non-chess club players, where the program plays in 'blitz' move (15 sec/move) against unlimited time for the human. In tournament play against humans, the program is currently winning about 50% of the games playing against players rated 1500-1700. The play of the pro gram is actually much better but various hardware problems and programming bugs have occurred to cause problems. In the last tournament entered, the program was substantially ahead in material in every game (at least a piece) but finally won two and lost two. In Tournament play against other programs the program has fared much better. It finished in a three-way tie for second place at the Seventh US Computer Chess Championship in Houston, Texas in October 1976 with a score of 3-1. It also qualified for participation in the Second World Computer Chess Championship held

in Toronto, Canada in August, 1977. The main reason for the strength of the program's play is the human-like approach to analyzing moves. These techniques allow the program to analyze a small tree of moves and search deeper than most programs and humans. In this way many traps and sacrifices are found that result in a material gain. *Blitz V* and *Duchess* became involved in a lengthy, time-consuming match as recounted below.

White - Duchess	Black - Blitz V
1. Pe2-e4	Pe7-e5
2. Ngl-f3	Nb8-c6
3. Bf1-b5	Bf8-c5
4. Pc2-c3	Pf7-f5
5. Pd2-d4	Pf5-e4:
6. Bd5-c6:	Pd 7-c6:
7. Nf3-e5:	Bc5-d6
8. 0-0 (a)	Ng8-f6 (b)
9. Bc1-g5	Bc8-e6
10. Nb1-d2	Ph7-h6
11. Bg5-f6:	Qd8-f6:
12. Qd1-h5+	Be6-f7
13. Ne5-f7:	Qf6-f7:
14. Qh5-g4 (c)	Bd6-f4
15. Nd2-e4:	0-0
16. Pg2-g3	Bf4-d6
17. Ne4-d6: (d)	Pc7-d6:
18. Pb2-b3	Ra8-e8
19. Pc3-c4 (e)	Re8-e6
20. Pd4-d5	Pc6-d5:
21. Pc4-d5:	Re6-e5
22. Qg4-d4	Pb7-b6
23. Ra1-d1 (f)	Rf8-e8
24. Qd4-c4	Re5-e2
25. Qc4-c6	Qf7-f6
26. Pa2-a3	Qf6-g6
27. Qc6-a4	Re8-e7
28. Qa4-c4	Re2-c2
29. Qc4-b4	Re7-e4
30. Rd1-d4	Re4-e2
31. Rd4-f4	Rc2-d2
32. Qb4-c4	Rd2-c2
33. Qc4-a4	Rc2-c7
34. Qa4-d4	Rc7-c2
35. Rf4-g4	Qg6-f7
36. Pb3-b4	Rc2-d2
37. Qd-4-c3	Rd2-d5:
38. Rg4-f4	Qf7-e7
39. Qc3-c6	Rd5-d2
40. Qc6-a8+	Kg8-h7
41. Rf4-f8	Rd2-a2
42. Qa8-f3	Qe7-e5
43. Rf8-d8	Ra2-d2
44. Qf3-f7	Pa7-a5
45. Qf7-g8+	Kh7-g6
46. Pb4-a5:	Pb6-a5:
47. Rd8-d7	Qe5-f6 (g)
48. Rd7-a7	Qf6-e5
49. Qg8-a8	Qe5-d4
50. Qa8-b7 (h)	Re2-f2:
51. Qb7-g7:+	Qd4-g7:
52. Ra7-g7:+ (i)	Kg6-g7:
53. Rf1-f2:	Rd2-d1+
54. Kg1-g2	Rd1-d4
55. Rf2-e2 (j)	

Annotations by Warren Porter

a) MCO suggests 8 Q-R5+ P-N3 9 Q-K2 Q-R5 10 P-KR3 B-K3 11 N-Q2 BxN 12 PxB Q-N4 13 0-0 BxKRP 14 QxP B-B4 15 Q-B3 0-0-0=

b) 8 . . . Q-R5 gives Black better chances – MCO. Now he can't support the KP.

c) White would trade queens since he would have no problems going into the endgame. As it is Black will have compensation for the pawn with pressure on f2 and command of the e-file.

d) I like 17 b3 here followed by Rae1. The knight's protection of f2 would give White time to challenge the e-file.

e) If 19 f4 to relieve pressure on f2, 19 . . Re3 20 Raci Rfe8 and Black will monopolize the e-file. White opens other lines for his Ra1 but will obtain a weak d-pawn in return.

f) White could defend the d pawn . . . this time.

g) At this point Black should look into the idea of h5-h4xg3. Blitz, which automatically looks at all captures, would certainly have found the mate in 4 if White should ever play h3?? or h4??.

h) Only move but good enough. Both machines and spectators had been looking at the tactical melee which follows. Instead of Black's next move it might still not be too late for . . h5 followed by either . . Kh6 or . . h4.

i) Fortunately this is check, otherwise Black has a mate in three.

j) This was about 12:45 am and I went to bed after *DUCHESS* printed the game history for me. The rest of the game was 55 . . . K-B3 56 R-K8 R-Q6 57 R-KR8 R-Q7+ (. . K-N2 and . . RxP draws easily.) 58 K-R3 K-N3 59 R-R8 R0Q4 60 R-K8 R-R4+ 61 K-N4 R-N4+ (61 . . RxP 62 R-K6=) 62 K-B4 K-B3 63 P-N4 R-QB4 64 R-K4 P-Q4 65 R-R4 R-B5+ 66 RxR PxR 67 K-K3? (Beat is 67 P-KR4 P-B6 68 K-K3 K-K4 and after the BK mops up the K-side he can still get back to c8 in time to draw. Black can win here with 67 . . K-N4! and after both kings have wiped out their opponent's pawn majorities and stepped aside, Black's h pawn will promote, preventing White from Queening on a8.) 67 . . K-K4? 68 P-QR4 69 P-R4 K-B4?? (Should lose to 70 P-N5 PxP 71 PxP K-Q4 72 P-N6 K-K3 73 K-Q4 K-B3 (. . P-B6 saves no time) 74 KxP KxP 75 K-N5 K-B3 76 KxP K-K2 77 K-N6 K-Q2 78 K-N7+ or 71 . . K-N5 72 P-N6 P-B6 73 K-Q3 K-N6 & White queens with check. Best is 69 . . K-K4 and if 70 P-N5 Black will be able to get back to QB1 in time to draw.) 70 K-K4 K-N5 71 K-Q4? (loses a tempo and should lose the game. With 71 . . P-B6! 72 K-Q3 K-N6 Black's new queen could stop the KNP. Best is 71 P-N5 which transposes back into what turned out to be the game continuation after Black's next.) 71 . . K-N6? 72 P-N5 PxP 73 PxP P-B6 74 P-N6 P-B7 75 P-N7 P-B8/Q. At about 3 am the computer at Duke went down and the game would be resumed only if necessary. *Blitz* predicts 76 P-N8/Q+ KxP Q-R2+, but of course the easiest draw is 77 Q-B4+. Ole Miss beat *DUCHESS* the next day so this game was ½:½.

Our thanks to Robert Hyatt, Ken Breland, Mike Penny and the rest of the USM CC for their help in this article and preparations for the 77 Miss. Championship.

Like any player you may meet at a rated tournament, *Blitz* has strengths and weaknesses. Treat it with respect and heads up play and your chance to score against it is greatly improved. – Warren Porter

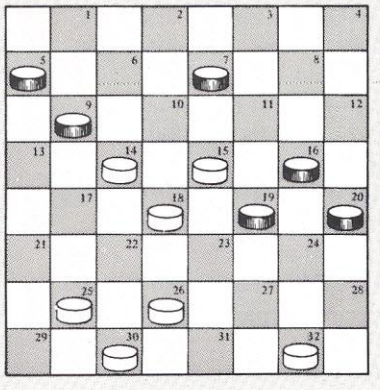
Computer Checkers

. . . . *Willard E. Matheson,* writing on "Brain and Machine" (*Personal Computing,* April 1978) describes Arthur Samuels' decision-tree routine of a checker-playing computer of the late 1950s. The number of possible moves to explore in a game of checkers is on the order of 10^{40} which is more than a billion times the number which expresses the radius of the universe in centimeters. The routine is similar to that employed in computer chess. The computer maps the checkerboard with a 32-bit word using four words to store a given board situation and to keep track of the different pieces. Examining each possible move and counter-move in decision-tree progression down to a given level, the machine compares the paths to find the optimum succession of moves. To do this the computer must assign to each move a relative value which takes into account the positional strength of each move. Although its program endowed the machine with a route learning that permitted it to remember past successes and to use them to advantage in new games the play of the machine was undistinguished. Subsequently the program was altered to instruct the computer to vary the weighting factors and remember the favorable results. This important program device enabled the computer to learn from its experience and to improve the quality of its play with successive games.

Duke beats Stanford at Checkers

. . . . Burke Grandjean of the American Checkers Federation sent in a story he had found buried in some papers. The story reported that Duke University defeated Stanford University in a computer-to-computer match at checkers in a brief match last summer. Duke supposedly then claimed the world title for its computer program since the Stanford program was previously regarded as the world's best. Duke's program was written by Eric Jensen and Tom Truscott (of the chess program, Duchess); Stanford's by Dr. Arthur L. Samuel. The American Checker Federation had cooperated with Dr. Samuel in the early 60's with his machine intelligence research; subsequently, Ken Hanson had assisted Dr. Samuel. Dr. Samuel wrote W.B. Grandjean sending along a copy of the games and requesting expert annotation from the ACF; he also pointed out differences between the two university programs. Following is one of the games as annotated by ACF Games Editor Dick Fortman. (The programs plyed "1-move restriction"; that is, the opening Black move was given, after the programs were GAYP (Go As You Please) actually a few book moves of the opening were available for the computers to follow.)

Black - Stanford	White - Duke
1. 11-16	22-18
2. 10-14(a)	24-19(b)
3. 8-11	25-22(c)
4. 7-10(d)	27-24(e)
5. 16-20	31-27(f)
6. 11-16	19-15(g)
7. 10x19	24x15
8. 6-10(h)	15x6
9. 1x10	28-24(i)
10. 4-8	24-19(j)
11. 3-7!!!(k)	18-15(l)
12. 9-13(m)	15x6
13. 2x9	29-25
14. 8-11	22-18
15. 13-17	19-15
16. 23-19	14x23
17. 27-18	12x19
18. 21-14	11-16(n)



Computer Checkers game. Black-Stanford. White-Duke. Position after 18th move. Black to play.

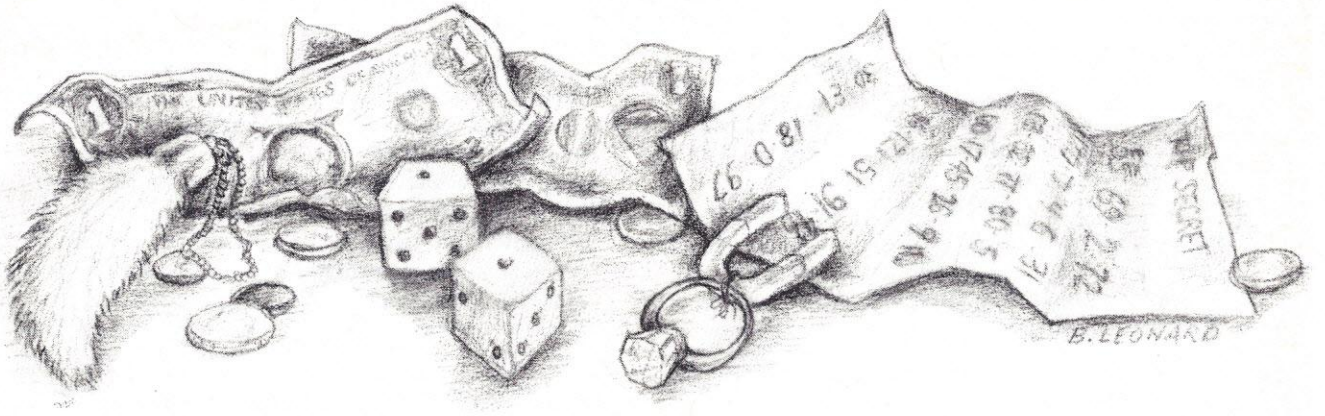
The game continued for 20 or so more moves, then Dr. Samuel conceded for Stanford. White played sloppily after the break-up, but with 2 and 3 pieces up, there is no need for precision. -- WPG

Dick Fortman's observations and annotations

a) Now the 3-mover, 10-14, 22-18, 11-16.
b) Satisfactory. Many prefer 26-22, 7-10, 22-12.
c) 26-22 preferred, then 16-20, 22-17, 7-10.
d) Good – 16-20 allows either 19-15 or 22-17.
e) 30-25 best, then 16-20, 22-17.
f) Draws but gives Black the strong attack. Instead, 19-16, 12-19, 23-7, 14-23, 26-19, 20-27, 31-24, 2-11, 24-20, etc., to draw.
g) Correct, 29-25 is a published loss after 3-7, 19-15, etc.
h) Also correct.
i) This is a probable White loss.
j) 29-25 offers more resistance.
k) Stanford's computer stubs an electronic toe here, transposing a win into a loss. Black should press against the White double corner beginning with 8-11.
l) Correct, as 29-25 allows the 9-13 exchange to draw.
m) 2-6 would put up more of a fight.
n) Resignation is now in order but perhaps computer programming does not cover this.

While many computer enthusiasts use only BASIC, some hobbyists prefer programming in machine language. Here, then, is a random number generator written in 8080 machine language.

The Random Way to Lose Your Dough



Generating random numbers is a relatively old activity in the history of civilization, dating back at least to the time when dice-throwing became a popular gamblers' game. Throwing a pair of dice in ten successive rolls produces a set of ten random two-digit numbers that have no relation to height, eye color, birthday or other features of dice throwers.

Some hardened gamblers, however,, claim they have ways of controlling the dice; rabbit's foot, crossed fingers, breath holding, yelling and cursing are some of their tricks. But the house man knows these antics will guarantee that the better will still lose his shirt.

Las Vegas knows you just can't beat random numbers. The most profitable game at casinos is the roulette wheel – provided no magnets have been attached to the underside of a woman's diamond ring.

The most dramatic use of random numbers occurred in World War II when super-secret codes were developed. Agents were given identical pages of random numbers. Every day a new random number was generated and transmitted to the agents who then used that particular page number to decipher or send intelligence.

Such a code is unbreakable within the short period it is used. By the time someone figures it out, a different code is already in use. □

EXPLANATION

This generator uses four addresses for temporary storage; they are 000,045; 000,046; 000,047; and 000,050.

ADDRESS		OP-CODE
000,000	041	LXI H/L
000,001	050	
000,002	000	
000,003	006	MVI B
000,004	010	binary 8
000,005	176	MOV A,M
000,006	007	RLC
000,007	007	RLC
000,010	007	RLC
000,011	256	XRA M
000,012	027	RAL
000,013	027	RAL
000,014	055	DCR L
000,015	055	DCR L
000,016	055	DCR L
000,017	176	MOV A,M
000,020	027	RAL
000,021	167	MOV M,A
000,022	054	INR L
000,023	176	MOV A,M
000,024	027	RAL
000,025	167	MOV M,A
000,026	054	INR L
000,027	176	MOV A,M
000,030	027	RAL
000,031	167	MOV M,A
000,032	054	INR L
000,033	176	MOV A,M
000,034	027	RAL
000,035	167	MOV M,A
000,036	005	DCR B
000,037	302	JNZ
000,040	006	
000,041	000	
000,042	XXX	Jump back to program or return.
000,043	XXX	
000,044	XXX	

How to Write for Personal Computing

Have you programmed your computer to converse in Gaelic? to do your home-ec homework? to read a bedtime story to the kids? Are you a frustrated fiction writer who's caught the computer bug? Or, have you found the ideal system or the absolutely worst combination of components?

Why not share your experiences with our readers? Yes, you too can write for *Personal Computing.* You choose the topic, *any* topic. If your topic relates to computers, great. If it relates to personal computers, even better. Computer hobbyists are looking for an excuse, any excuse, to buy a computer, and you might just offer the justification they're looking for.

We accept articles for all our sections – *Launching Pad* (our tutorial section for beginners), *On the Lighter Side* (where we print humorous applications), *In the Money* (how to use your computer to benefit financially), *Digging In* (for our more "advanced" topics), and *Once Upon a Time* (where we let your imagination run wild). We'd love to see some comparisons of computers or computer products. Tell us the good *and* bad of your system.

Keep your writing simple. No, our readers are not simpletons or beginners, but if you can explain something in simple words, do so. Don't clutter your piece with unnecessary jargon. If you're already into computers, give the newcomers a hand and let them in on some of the tricks of the trade – in simple terms. Examples, analogies, and charts and diagrams help both the beginner and the more advanced user appreciate what you're saying. Feel free to use "I" and "you" to make your article more personal and meaningful to the reader. Put the reader in the position of programmer ("you"). Also, please do not write your entire article in caps. And please indent for each paragraph.

Some things to note. Make sure your details are accurate – especially prices, other numerical information, and company names. Don't rely on hearsay or memory.

If you write about a program you've invented, try this order (to make sure you cover all angles): state the program's purpose; show a sample run; explain what the input options are, and what the output means; show another sample run; explain the underlying theory (if any); state the language, version, and computer you used and their peculiarities; show the listing; explain the program's over-all structure; analyze the program's details line by line; and suggest how the reader might improve or change the program.

Whatever your area of interest, you can turn it into an article. For example, if you're interested in watching birds then why not try an article on how to use a computer to track bird migrations? Or if your business is _____, why not try a piece on computers and how they can be used to _____ _____. We're open to ideas . . . .

If you've never written for a publication before and you'd like to discuss your piece with us before beginning it, give us a call. (Please do *not* mail us vague story proposals or outlines. We'd rather see the first few paragraphs of your article.) We'd be glad to discuss what you have in mind, and offer a few ideas of our own

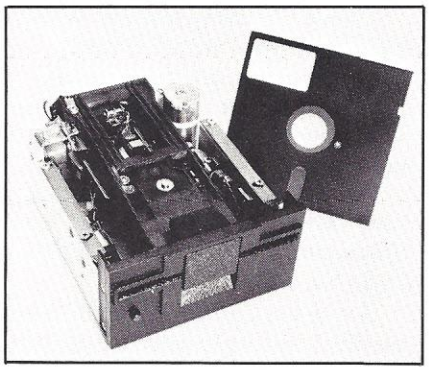
As a matter of form, we prefer (and are more likely to accept) articles that have been typed. Most of our articles run around 2-4 magazine pages. (There's about 3-3/4 typewritten pages to a magazine page.)

Now here's the good part: we pay for any original material we print, although the price varies depending on the *quality* of the article. (So make it good!)

Why not give it a whirl? There may be a latent Hemingway, Fitzgerald or Asimov beneath that Einsteinian exterior.

Systems, Subsystems, Software

A double-head version of the double-density FD200 Microfloppy™ disk drive has been introduced by Pertec Computer Corporation's Pertec Division, Chatsworth, CA. The new **FD250**



Microfloppy Disk Drive stores up to 437,500 bytes without operator intervention (i.e., without flipping the 5.25-inch diskette). Double-density, hard or soft sectoring, and write protect are all standard features. The unit can write and read data on both sides of a diskette.

The new disk drives will be manufactured and marketed on an OEM basis by Pertec Division, and will be incorporated into MITS/Altair™ and iCOM® products manufactured and marketed by Pertec Computer Corporation's Microsystems Division.

Measuring 3.25 inches in height, 5.75 in width and 8 inches in depth, the FD250 weighs 3.2 lbs. Its seek time is 25 msec. track-to-track, with head settling time of 10 msec. (last track addressed) and head loading time of 35 msec. (maximum).

Recommended recording mode is FM on single-density, and MFM on double-density. Recording density (inside track) is 2768/5536 bpi, with 1,750,000 bits per disk (double-density) [unformatted].

For further information, contact Neil McElwee, PCC, Pertec Division, 9600 Irondale, Chatsworth, CA 91311; (213) 822-9222. Circle No. 101.

A complete and inexpensive S100 Power Supply is now available from GGRS Microtech. This supply provides unregulated +8 VDC @ 10 Amps as well as +16,-16 VDC @ 1 Amp. The S100 Power Supply kit comes with special transformer, rectifiers, capacitors, mounting hardware, and metal foundation. This supply sells for $49.95 and further information is available from CGRS Microtech, PO Box 368, Southampton, PA 18966; (215) 757-0284. Circle No. 103.

Instant Image is the name of an innovative computer portrait company; and that's what you get from their computer portrait system.

The compact system consists of a television camera and monitor, computer and line printer. The printer is programmed to use characters of varying densities corresponding to the patterns of light and dark produced by the TV camera. A 12" by 12" portrait consisting of up to 14,400 characters is completed in 60 seconds.

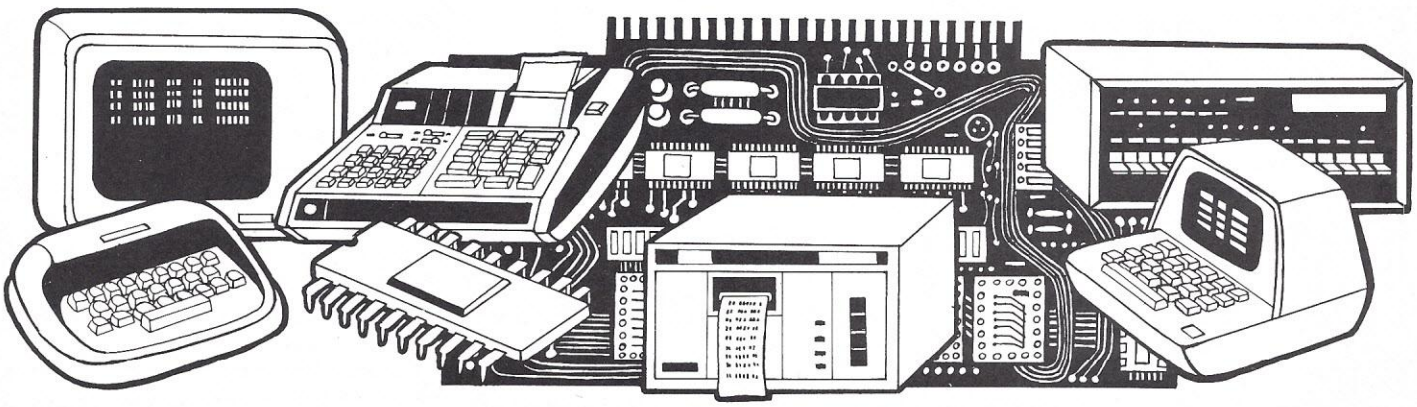
INSTANT IMAGE has also designed long-term programs for supermarket chains and savings and loan associations for use at the consumer level.

The portraits can be permanently transferred to T-shirts and many other promotional items giving them a unique, personal touch.

The INSTANT IMAGE people have extensive experience in marketing and sales promotion and work with each

client in developing individual programs to maximize the success of their promotion or exhibit.

For further information regarding INSTANT IMAGE computer portrait systems, contact Sandie Johnson at INSTANT IMAGE, 1651 East Edinger Avenue, Suite 102, Santa Ana, CA 92705; (714) 558-7965. Circle No. 105.



The 1550 Stringz 'n' Thingz kit is the latest polytonic keyboard system from PAIA Electronics. This multi-purpose instrument is capable of generating orchestral textures consisting of violin, cello, and piano voicings.

A full complement of operator controls allows switch selectable keyboard split, separate mixers for upper and lower keyboard, variable vibrato/chorusing rate and depth controls to allow reed organ and pipe organ voicings, and variable sustain controls for piano and strings.

A standard gate trigger jack allows interfacing the 1550 to any of the commercial synthesizers which feature systems interfacing jacks. This allows capabilities for brass synthesis, filtered strings, and other polytonic synthesizer effects.

Optional features include foot pedals for volume or sustain time control, foot switches for sustain control, and the 1551 Stereo Option to convert the mono string output to a true stereo output with two switch selectable modes of stereo operation.

The other options include a microprocessor interface to allow "memorization" of string or piano parts for later reproduction at any tempo and key desired. Also, the 'processor interface' will allow the 1550 keyboard to simultaneously control a modular polyphonic synthesizer system.

The complete Stringz 'n' Thingz kit including 84 page, well-illustrated, step step-by-step assembly and operation

manual is available direct from PAIA for $295.00 (plus shipping, 30 lbs.) and through select retail outlets. Custom, factory assembly available; dealer inquiries invited.

For more information contact: PAIA Electronics, Inc., 1020 W. Wilshire Blvd Blvd., Oklahoma City, 73116; (405) 842-5480. Circle No.106.

A 6502 Microcomputer Card for the S100 Bus structure is now available from CGRS Microtech. OEMs and end users who wish to use the high speed of the 6502 microprocessor will find its 1MHz crystal clock a significant feature.

The board contains 2K of RAM and 4K of 2708 EPROM. Kit price (less memory) is $149.95 and assembled price (less memory) is $189.95.

Among the many features of the 6502 MPU board are an on-card voltage regulator, power-up restart circuitry, a 50 pin front panel connector, slow memory interface logic, and S100 interface logic.

For more information about this write CGRS Microtech, PO Box 368, Southampton, PA; (215) 757-0284. Circle No. 140.

A new low-cost graphics CRT terminal that is programmable in a high-level language and offers ease of operation and flexibility in graphics applications is Hewlett-Packard's entry into the intelligent terminals market. The top-of-the-line HP2647A graphics terminal, which also offers full interactive alphanumeric capability, features multiple display workspaces, shares output peripherals, displays data as graphs, pie or bar charts and provides dot-by-dot hard hardcopy of its screen display with optional companion plotter/printers.

Using a subset of HP BASIC in the HP 2647A raster-scan graphics terminal, the unit's operational characteristics can be tailored to meet specific needs of users to solve a variety of problems in engineering, scientific and business environments.

By sharing intelligence with the microprocessor-controlled terminal, a host computer's resources are freed for more complicated tasks saving computer time and communications costs. The terminal accepts BASIC programs that are downloaded from the host CPU and then executes them under local control. With BASIC, the terminal's graphics and alphanumeric functions and facilities can be modified, output from a computer can be changed into formats defined by the user and the keyboard can be reconfigured by assigning each key a different code. With such flexibility, no software changes may be required to adopt "canned" programs to users' applications.

Multiple workspaces enable the terminal's display memory to be partitioned into four separate sections. These can be varied in size using dynamic memory allocation and used for such tasks as conducting an interactive session with the computer, displaying an output of an applications program, showing a local BASIC program and acting as a scratch pad for computation. When computation is needed during an interactive session with a host computer, the user can switch to another workspace, perform the calculation through several BASIC statements, and then switch back to enter the results.

English-like commands from a host computer or a locally-entered BASIC program may be used to control the graphics features of the HP 2647A. Commands like PLOT, AXES, LOCATE, DRAW and LABEL are an extension of BASIC that give users an easy to use set of instructions designed to minimize the costly and time-consuming process of preparing and implementing graphics programs.

Another feature that minimizes software investment is that the terminal can generate multiple automatic plots from either horizontal or vertical columnar data. A feature that is independent of system and software, automatic plotting enables users to generate fully labelled pie charts, bar charts and X-Y Cartesian graphs with a few keystrokes. Different types of shading are available to distinguish various chart areas. Little or no programming skill is needed to plot data since the user is guided through a simple menu of key questions about the data. Data to be plotted may be from the host computer, from the terminal keyboard or from the standard integrated dual cartridge tape drives that provide a total of 220K bytes of local data storage.

To control data paths among terminal facilities and peripherals the user is guided by a command display at the bottom of the screen which labels the functions of the eight command keys. The same keys also serve as softkeys whose user-assigned functions are also displayed in an eight-character format to remind the user of their content.

The new HP 2647A is compatible with programs developed for the HP 2648A – HP's first graphics terminal. It offers all the capabilities of the earlier terminal including independent graphics and alphanumeric memories, a bright display of 360 x 720 individually addressable points, selective erase, system-independent zooming and panning, and rubber-band line drawing that can be used without CPU support.

In addition to these features, the intelligent graphics terminal gives users all the capabilities of the HP 2645A display terminal including data communications flexibility, user-defined softkeys, off-line data preparation and editing capability, microprocessor-controlled memory allocation, built-in self test and both page and character-mode operation. The unit's 5" x 10" high-resolution display presents data in a 24-line by 80-column format and provides clear and easy-to-read alphanumeric characters using 9 x 15-dot character cells.

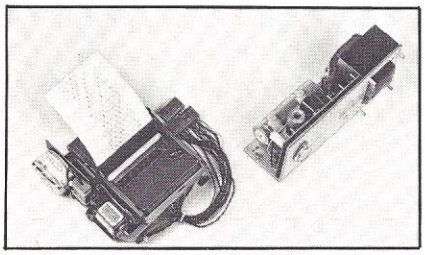
The terminal has 64K bytes of RAM for BASIC, 32K bytes of RAM for graphics and 56K bytes of ROM for terminal control functions.

High-quality, vector-drawn hardcopy for the terminal can be provided by both the HP 9872A multi-color graphics plotter and the HP 7245A plotter/printer. A newly introduced option to the HP 7245A enables users to get a dot-by-dot hardcopy of the HP 2647A graphics display memory. Price of the HP 9872A is $4,200, while the HP 7245A is priced at $4,600. The new option adds $250 to the plotter/printer's base price. (U.S. prices only).

The Hewlett-Packard 2647A intelligent graphics terminal is priced at

$8,300 (U.S. price). First deliveries are scheduled for July. For more information, contact Inquiries Manager, Hewlett-Packard Company, 1507 Page Mill Road, Palo Alto, California 94304. Circle No. 110 .

A family of **impact print mechanisms**, the associated controls, and power supplies for inexpensive column printing have been introduced by SODECO, A Division of Landis & Gyr. Systems elements include impact printing mechanisms for 15 and 21 columns, an interface/controller based on Mostek 3870 microprocessor, an interface/controller based on F8 microprocessor, system power supply, and separate programmed 3870 "chips."



Commenting on the availability of this low-cost impact printing system, Peter M. Engstrom, Marketing Manager, said: "For the first time the variety of low-cost computer and microcomputer applications can enjoy the versatility of a hard-copy printer that is compatible with the overall system cost. Indeed, for under $300, an OEM can purchase all the system building blocks he may need. That is, for this money he will get a *complete* impact printing system capable of accepting ASCII (serial or parallel), RS232C, or BCD data formats and printing it at 1.5 seconds/line in the case of an alphanumeric printer. We find this system in a wide variety of data logging, POS (point-of-sale), banking, medical and other applications."

Sodeco offers Series PR15 and Series PR21 printing mechanisms, that can print either 15 or 21 columns, respectively, numeric only or full alphanumeric models or 1.5 lines per second for full alphanumeric. These speeds are due in part to the use of a single hammer to form characters for three adjacent columns, rather than a hammer for each column. Not only does this increase printing speeds, but it also raises the reliability by reducing the number of moving parts and lowers the cost of solenoid driver circuits.

Measuring 5.8" across by 6.0" deep by 2.3" high, the PR Series printers use snap-in two-color ribbon cartridges for quick and effortless ribbon changing. The second color permits highlighting more important data or messages.

Prices in quantities of 100 or more are $105 for the 15-column printer and $130 for the 21-column model.

To simplify interfacing of Series PR printers with various common data sources Sodeco offers two interface/controllers. One of them, Model 4-621-9210, uses a Mostek 3870 microprocessor.

Both interfaces accept ASCII (serial and 8-bit parallel, RS232C or BCD parallel four bits) data data formats. Both interfaces contain all the circuitry required to operate the printers, including the hammer solenoid drivers, ROM character generator, a full line buffer, timing control, full "handshaking" facilities, and related logic necessary for interfacing and controlling the Series PR printer.

Prices for the 3870-based interface board are $120 in lots of 1 through 9 and $250 for the F8-based board. In addition, the 3870 "chip" – completely programmed – is available separately for $40 in unit quantities.

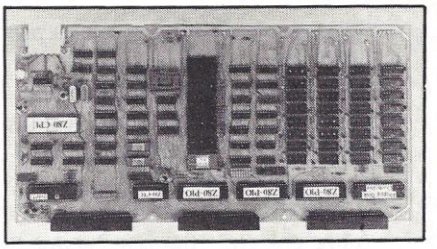
To provide power for the above systems, Sodeco offers a custom power supply that has all the voltages required by the Series PR15 and 21-column printers and the associated interfaces. Designated as "Model CP242", the unit is small (2-3/4" x 3-1/4") and can operate either on a 120 Vac, 60 Hz or 220 Vac, 50 Hz line.

Prices for the CP242 are $90 in unit quantities and $70 in lots of 100 or more. For more information contact Sodeco, Div. of Landis & Gyr, 4 Westchester Plaza, Elmsford, NY 10523; (914) 592-4400.

The Small Computer Buying Guide in two volumes (one for hardware, the other for software) will be available this fall in heavy-duty soft cover. Introductory data, product evaluation charts and detailed reviews included. Circle 125.

A Z80 based, single board microcomputer with resident floppy controller has been released by Quay Corporation, P.O. Box 386, New Jersey 07728. The **90F/MPS microcomputer,** intended for OEM applications, supports up to four 5-1/4" or 8" single density, floppy drives.

In addition to the floppy controller, the basic board includes: 4K dynamic



RAM (expandable to 64K), six 2708/2716 EPROM sockets, 1K PROM resident Monitor with static RAM scratch pad, (1K), PROM Programmer, Z80-Counter/Timer chip, hardware UART with RS23C/TTY serial I/O), and a programmable 8-bit parallel I/O port with expansion sockets for two additional Z80-PIO chips (16 lines each).

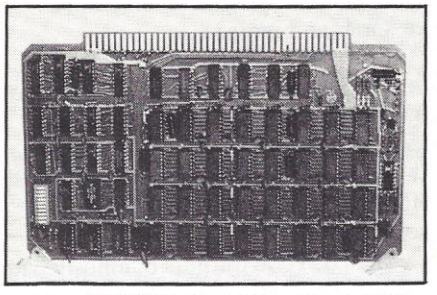
Single unit price for the 90F/MPS is $995, with OEM discounts available. Product availability is 30-60 days ARO. Factory may be contacted for further information. Circle No. 114.

Designed specifically for operation with Motorala EXORcisor® and MEC 6800 evaluation module is Chrislin Industries' new C16800 16K x 8 semiconductor memory system. The new memory features easy expansion to 32K, 48K or 64K equivalent with no further modification to the board. **CI6800 memory board** plugs directly into existing EXORcisor® Connectors.

The CI6800 allows maximum processor throughput with the use of hidden refresh control logic on board. Data access time is 300 nsec and cycle time is 750 nsec.

On board memory select is available in 4K increments up to 64K words of memory. Write disable switch on board makes the RAM appear to be a ROM.

Complete board power consumption is under 5 watts. Size is 5.75". OEM



pricing $390.00 for 16K x 8 and $99 $995.00 for 64K x 8. Delivery is 2 - 3 weeks. For more information contact Chrislin Industries, 31312 Via Colinas, 102, Westlake Village, CA 91361; (213) 991-2254. Circle No. 115.

Micon Industries, Oakland, CA, manufacturer of low cost data terminals, announces their first hard copy data communications terminal. Named PERT **(Printing Economy Remote Terminal)**, the new device provides 48 characters per line of printout on a continuous roll of ribbonless thermal paper. According to Micon Industries president William Northfield, "The modular design of PERT allows the customer a large choice of options including rechargeable battery or 110 VAC power, line level or acoustic coupler telephone line connect, direct computer interconnect with a 20 mA loop or RS232."

Features of the new PERT include 110-9600 baud selectable data transfer rates, standard ASCII upper case keyboard with all necessary characters and status indicator lights, full

compatibility with all standard computer formats and all standard US and Canadian telephone equipment. PERT is available in eight color options and is covered by a full year warranty on materials and workmanship.

PERT joins a full line of compact, low-cost Micon data terminals which include MIGET (Miniature Interface General-purpose Economy Terminal), Pocketerm (a battery opperated telephone interface economy terminal) and Cassetterm (a telephone interface terminal with mini-cassette storage.) All Micon terminals are available through a special mail order/factory direct purchase system on a ten day free trial basis. For more information, contact Micon Industries, 252 Oak St., Oakland, CA 94607; (415) 763-6033. Circle No. 116.

A complete and inexpensive S100 Power Supply is now available from GGRS Microtech. This supply provides unregulated +8 VDC @ 10 Amps as well as +16,-16 @ 1 Amp. The S100 Power

Supply kit comes with special trans-orformer, rectifiers, capacitors, mounting hardware, and metal foundation. This supply sells for $49.95 and further information is available from CGRS Microtech, PO Box 368, Southampton, PA 18966; (215) 757-0284.

Robins **Floppy Disk Desktop Stand** is designed for desktop or wall mounting. It holds 20 diskettes in side access shock resistant pockets. The unit is supplied with 20 diskette pockets and color coded indexes and axles. Additional stands increase filing capacity.

The Floppy Disk Desktop Stand, Robins catalog #78-051, carries a professional net price of $64.

For further information contact Samuel Jones, Sales Manager, Robins Industries Corp., Data Products Division, 75 Austin Blvd., Commack, NY 11725; (516) 543-5200. Circle No. 117.

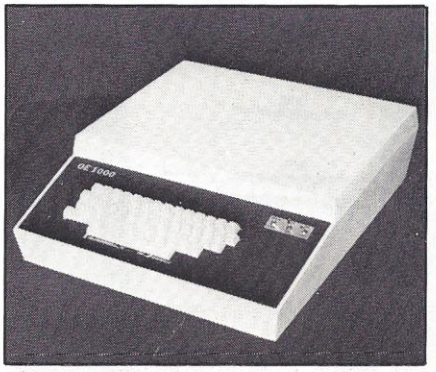
A big, colorful 104-page catalog that describes the newest in electronic kits for the do-it yourselfer is now available from Heath Company, Dept. 350-660, Benton Harbor, MI 49022. Product categories in the new catalog include amateur radio, high-fidelity components, color television, test instruments, digital clocks, auto, marine and aircraft accessories, electronics learning programs, personal computer systems and much more.

Products new to the **summer catalog** include a floppy disk for the H8 Computer System, a fully-assembled and tested H11 Computer and accessories, a new small appliance touch control switch, an ultrasonic auto burglar alarm, amateur radio antennas and accessories and more.

Circle No. 119.

The **OE1000 terminal** is designed to interface to any microcomputer that has a 300 baud serial data output port. It operates in the full duplex mode with either 20 mA current loop or a RS232 voltage swing.

The OE 1000 outputs composite vid video for use with a modified TV or video monitor. The screen format is 16 lines by 64 characters. It has an upper and lower case mode or TTY mode key keyboard and will display 96 ASCII characters and 32 special characters. The OE 1000 has full cursor control,

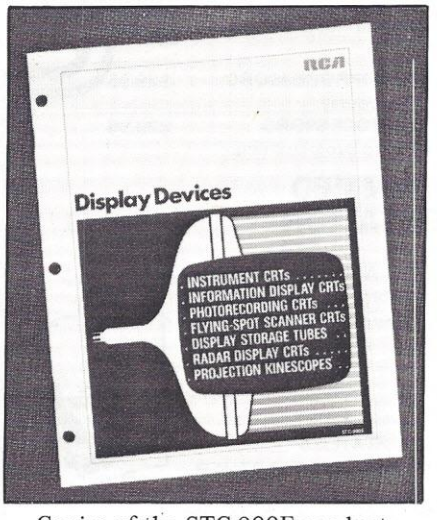


automatic scroll, erase to end of line, erase to end of screen, and clear screen.

The OE 1000 terminal may be ordered, as a kit for $275 or assembled for $350, from OTTO ELECTRONICS, P. O. Box 3066, Princeton, NJ 08540, (609) 448-9165. Delivery in 2 weeks. Dealer inquiries invited. Circle No. 120.

A thirty-eight page product guide providing tabulated data and outline configurations for RCA's standard line of display devices has been released by RCA Electro-Optics and Devices, Lancaster, Pennsylvania.

The **product guide, STC-900E**, features a comprehensive treatment of display devices, terms and definitions and phosphor screen characteristics. The guide includes data on the following types of display devices: instrument CRTs, information display CRTs, photorecording CRTs, flying-spot scanner CRTs, display storage tubes, radar display CRTs, and projection kinescopes.



Copies of the STC-900E product guide may be obtained by writing to RCA, Box 3200, Somerville, New Jersey 08876, or by calling (717) 397-2712. Circle No. 121.

ComputerCo, Inc, announces the release of its **KFAM system software for the 8080, 8085, and Z-80** microprocessor. Designed around North Star's BASIC, version 6 with 14 digit precision, the micro KFAM saves the engineer or programmer 80% of normal time required to code the original application program or modify an existing subsystem. With ComputerCo's KFAM, the necessary overhead of software expertise is greatly reduced.

The application programmer utilizes KFAM subroutines to handle data transmittal, packing and unpacking of

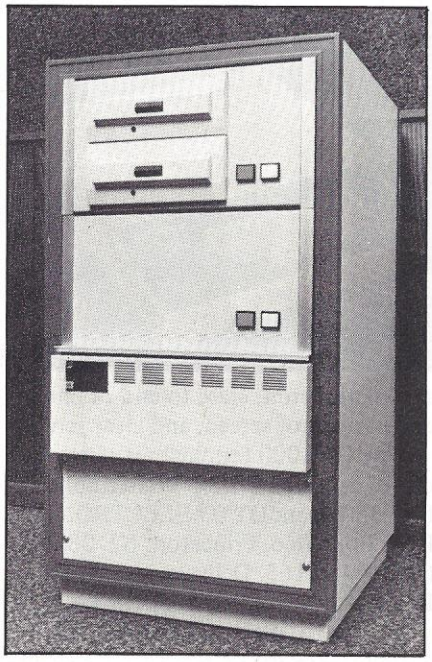
data to maximize storage area, sorting upon input by keys, opening and closing of files, self verifying of files, and the modular design of application software. KFAM contains a keyboard input utility for displaying data on a video screen, accepting the keyboard input, cursor positioning, and validation of data. Utilities to add records, delete records, and examine or alter existing records are included.

ComputerCo, Inc., offers the KFAM system software with complete documentation for $550. Documentation only without soruce code is $75. KFAM is available on diskette or tape. Application software is also available. Dealer inquiries welcomed. For more information contact ComputerCo, Inc., 5833 Dorchester Road, Charlston, SC 29405; (803) 552-8533. Circle No.126.

Ohio Scientific announces the C3-B; the first fully packaged Winchester disk based microcomputer system. The **Ohio Scientific C3-B** is a package microcomputer system in a 42" equipment rack. The system includes in its minimal configuration, 48K of static RAM, Ohio Scientific's triple processor CPU board which has 6502A, 6800 and a 74 million byte Winchester technology fixed disk.

The Winchester disk communicates with the CPU via a dedicated high speed memory channel which services a dual port memory. This unique architecture allows high performance disk operation with no degradation of processor speed or use of interrupts.

The C3-B features a 16 slot case in which only 7 slots are used in the base machine, allowing tremendous expansion including memory up to 768K, three additional Winchester disks for 300 megabytes on line storage and 16 communications ports. Because of the disks' intelligent controller, the CPU is completely available for communication such that it can always service terminal interrupts. This is very important for high throughout operation in multi-terminal configurations.



The C3-B system comes complete with OS-65U disk operating system with extended BASIC. This operating

system features virtual data files and directly supports high performance file structures such as multi-terminal applications software is now available for the C3-B.

The C3-B has been in production since February, 1978, and is available now on a reasonable delivery schedule. The single unit price is $11,090 with quantity discounts being available to qualified OEM users. For more information contact Ohio Scientific, 13333 S. Chillicothe Rd., Aurora, OH 44202; (216) 562-3101. Circle No. 112

A "de-jargonized" instruction manual has been developed by PolyMorphic Systems for its advanced **System 88 microcomputer** which, combined with the company's new BASIC language disk, enables any business or professional person to operate the computer himself for complex problem-solving and word processing.

"The System 88 is an important step forward in the evolution of microcomputers," said Richard Petersen, president of PolyMorphic. "Until now, people without special training were unable to use computers of such sophistication for their specialized business and professional needs.

"The system 88 has changed all that. While the applications of the system are almost unlimited, no technical knowledge is demanded of the operator if he follows our new plain-talk System 88 instruction manual."

An additional manual, **"System 88 Disk, BASIC,"** is provided for those with more advanced understanding of computers who wish to do their own programming.

System 88 microcomputers consist of a main unit about the size of a stereo tuner; keyboard, and video monitor – truly compact for the home or office.

The new system disk includes enhanced operating software, BASIC, text editor and assembler. Added to the BASIC language are string arrays and array commands, which allow the user to create and manipulate such items as mailing lists, personal records, inventory, and billing. For scientific use, BASIC now offers inverse trig, hyperbolic, and gamma functions.

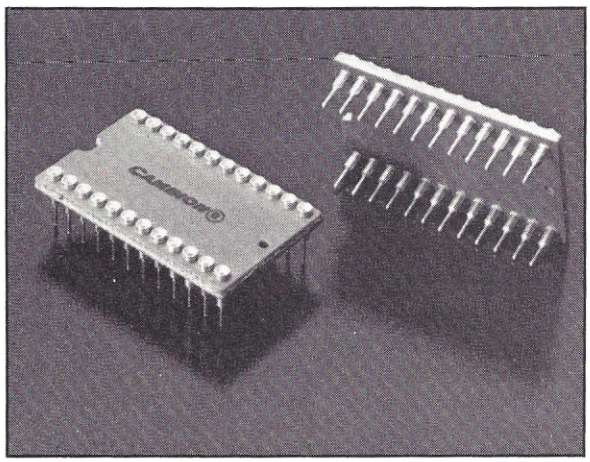
The improved text editor permits the user to move, duplicate, or delete entire blocks of text. This makes possible fast large-scale reorganization of text. Macro instructions have been added to the assembler.

Advanced Micro Devices of Sunnyvale, CA, now offers user-programmable, read-only memories, featuring proprietary fuse technology. These **AMD PROMs** offer fast programming platinum-silicide fuses claimed to have over two billion hours of failure-free life tests in AMD's laboratories. Platinum-silicide produces a clean and well-defined fuse that does not change during operation. Additionally, says the company, this proprietary fuse offers fast programming with excellent yields. Typically, a 4096-bit part can be programmed with a single 50 microsecond pulse in less than one second. These PROMs are all processed using the company's low-power Schottky technology, claims the spokesman for the firm. Three-micron thick epitaxial layers are combined with washed emitters, composite masking, dual-layer metallization and barrier metal Schottky diodes to produce these PROMs. All parts are said to maintain designated speed specification over full military temperature operating range. Prices for the 256-bit unit start at $195. For more information, contact Advanced Micro Devices, Inc., 901 Thompson Place, Sunnyvale, CA 94086. Circle No. 104.

The most recent addition to the Syntest Corporation Digital Printer line is the **SP-308 Ticket Printer Subsystem.** Designed for OEM use, this subsystem mates the power of the microprocessor and the LRC 7040T print mechanism to provide a method of ticket printing. The 40 column 5 x 7 dot matrix, impact printer accepts multi-copy forms up to 11" x 17" in size. A maximum of 22 lines may be printed at 50 characters per second. The microprocessor controller accepts ASCII data in either RS 232C or 20 mA current loop formats. Standard data rates to 9600 baud are available. Parity and number of stop bits is switch programmable. The SP-308 provides 40 character buffering, double width print capability, tab function and pressure roll release control. Additional features include jumper selectable print intensity, standard 25 pin EIA input connector, and printer busy signal.

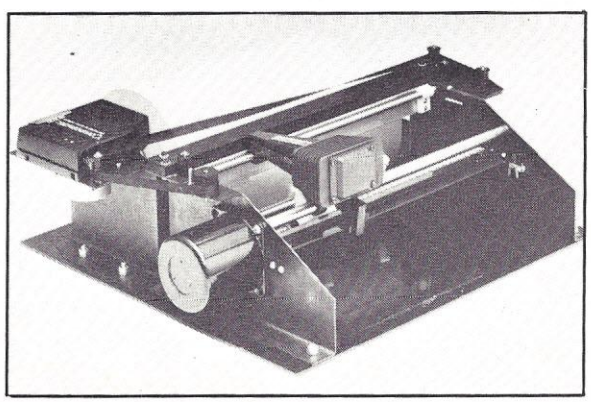
The SP-308 subsystem requires an installation area of 7.0" high x 8" deep x 8.5" wide. Power requirements are 110/220 VAC 50/60 Hz@25 watts. Price is $485 in 100 piece quantities. Delivery is 30 - 45 days. For more information, contact Syntest Corp., 169 Millham St., Marlborough, MA 01752. Circle 107.

A 24-pin socket that prevents damage from frequent unplugging has been developed by Cambridge Thermionic, Cambridge, MA. Called the **Cambion 24-pin PROM Protector Socket**, the appliance features contacts with screw machined bodies and beryllium copper contact inserts, all designed for durability. The sockets will accept most 24-pin DIP packages and pins from .014" to .020" diameter. These protector sockets are available in either of two platings: -03 (gold 50 micron.) or -04 (bright tin 200 micron).



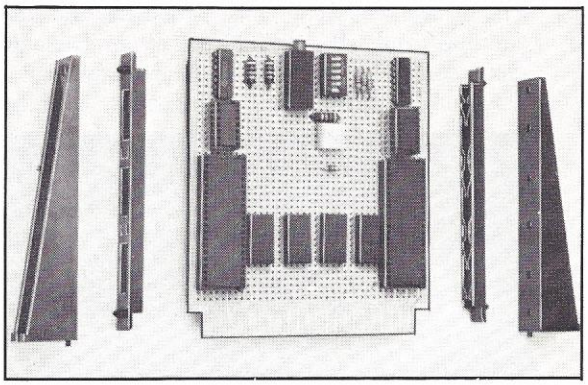
For more information, contact Cambridge Thermionic Corp., 445 Concord Avenue, Cambridge, MA 02138. Circle No. 102.

A new bi-directional **peripheral printer mechanism** has been designed by MarComm Inc. of Ramona, CA, for simplicity of operation and reliability. The print rate is 120 cps using the impact, dot matrix print technique. Line capacity is 80 characters at 10 characters per inch. This printer uses 8-1/2" standard roll paper, with up to 5 ply multiple copies and has a paper slew rate of 400 lines per minute. Paper feed is by a pressure roller, with motorized paper advance. The ribbon is in a cartridge, with independent power motor. There are 6 lines per inch. Power requirements: 117 VAC, 60 cycle. Opto isolators for BOL, EOL and character positioning and sensing. The printer mechanism includes interface electronics. For details write MarComm Inc., PO Box 535, Ramona, CA 92065; (714) 789-3833. Circle No. 109.



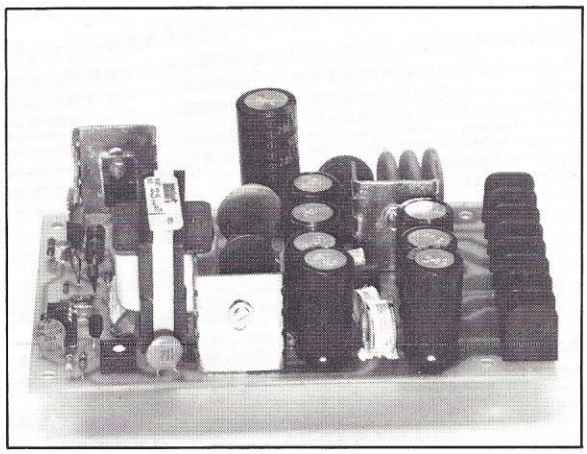
WHAT'S COMING UP

A new PC cardguide is available from O.K. Machine and Tool, Bronx, NY. **Cardguide Model TR-1** consists of two guides molded of black plastic. Spring-finger action is used to dampen shock and vibration while permitting smooth insertion or extraction. The new guides accommodate any card thick-



ness from .040 to .100 inches. Overall length is 4-1/8 inches. The guides feature snap-in buttons for easy assembly to rack or brackets and are priced at $1.89 per pair. For more information, contact O.K. Machine and Tool Corp., 3455 Conner St., Bronx, NY 10475. Circle No. 101.

Boschert, Inc. of Sunnyvale, CA, claims its new 25-watt quad-output power supply challenges other linear power supplies in terms of efficiency, size and weight. Designated the OL-25, the new "switcher" is aimed at such applications as small microprocessing systems, mini cathode-ray-tube display terminals, desk-top calculators, and almost any piece of elec-



tronic equipment requiring 25 watts. The **OL25 Switcher** operates at 60 to 70% efficiency and dissipates relatively little heat. Operating at 20,000 hertz, the unit allows use of smaller transformers, inductors, and filter capacitors. Its measurements are 2.5 x 4 x 6 inches and weighs about 12 oz. It is also claimed to be brownout proof. Price of the unit is $80 each in 100-unit quantities. For more information, contact Boschert, Inc., 384 Santa Trinita Avenue, Sunnyvale, CA 94086. Circle No. 105.

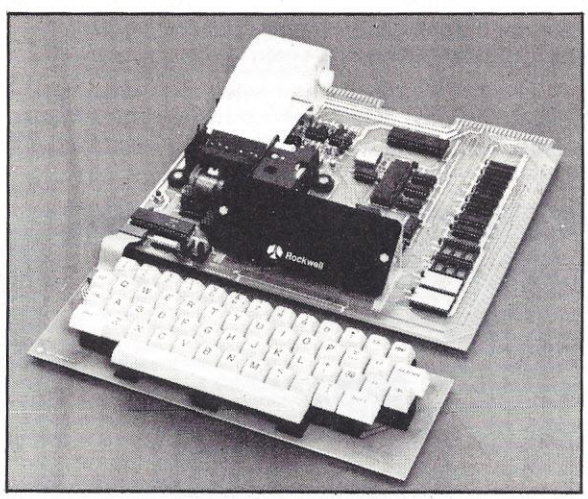
Rockwell International has introduced a new single-module microcomputer system featuring an on-board 20-column printer.

Designated **AIM 65 (R6500 Advanced Interface Module)** the new system is intended as an educational aid for first-time users and a general-purpose microcomputer for engineers. AIM 65 is also designed to be an effective, low-cost microcomputer development system.

AIM 65 features on-board alphanumeric 20-character printer and display and a 54-key terminal-style keyboard. Available in 1K- and 4K-byte RAM versions, AIM 65 is designed around the R6502 CPU, which has 64K address capability with 13 addressing modes and both decimal and binary functions.

An 8K ROM-resident monitor program provides all peripheral control and user programming functions. Spare sockets are included to further expand on-board program memory via user PROM-based programs or Rockwell's assembler, text editor and BASIC interpreter plug-in ROM options.

The AIM 65 board also has a connector that allows external access to the system bus for memory and I/O

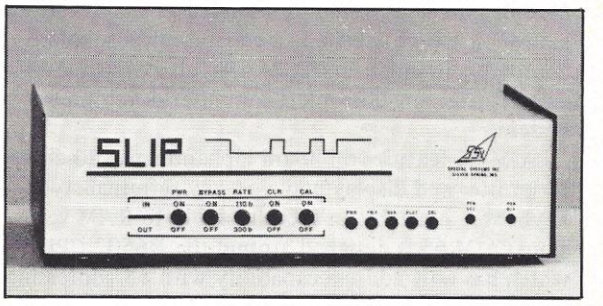


expansion. A separate application connector interfaces a TTY and two standard cassette recorders, and includes a user-dedicated Versatile Interface Adapter. The VIA features three 8-bit bidirectional ports (two parallel, one serial) and two 16-bit interval timer/event counters.

Rockwell is implementing a long term product development program stemming from the R6500 NMOS family as part of a "total microelectronic system solution commitment," according to Bob Anslow, director, LSI Products at Rockwell's Microelectronic Devices.

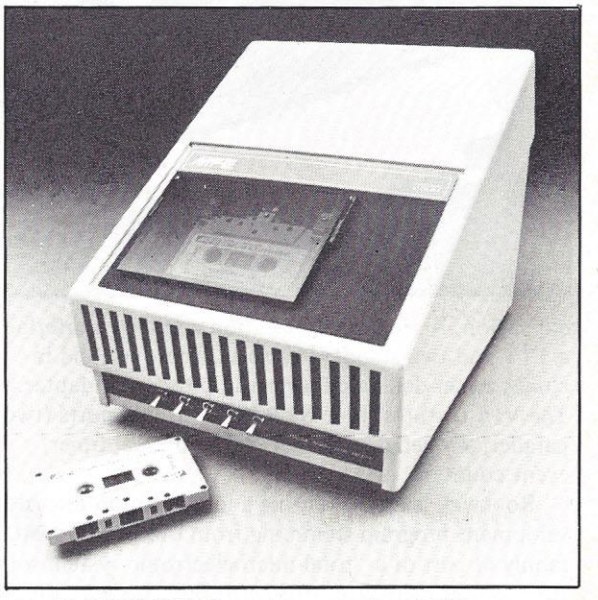
Rockwell International is a major, multi-industry company applying advanced technology to a wide range of products in its aerospace, automotive, consumer electronics, utility and industrial businesses. For more information, contact Rockwell International, Electronic Devices Division, PO Box 3669, Anaheim, CA 92803. Circle No. 108.

A new economical means for providing plotting and graphics capability in lab or office is claimed by Special Systems, Inc., Silver Springs, MD. Called **SLIP (Serial Language Independent Plotter) Controller**, the

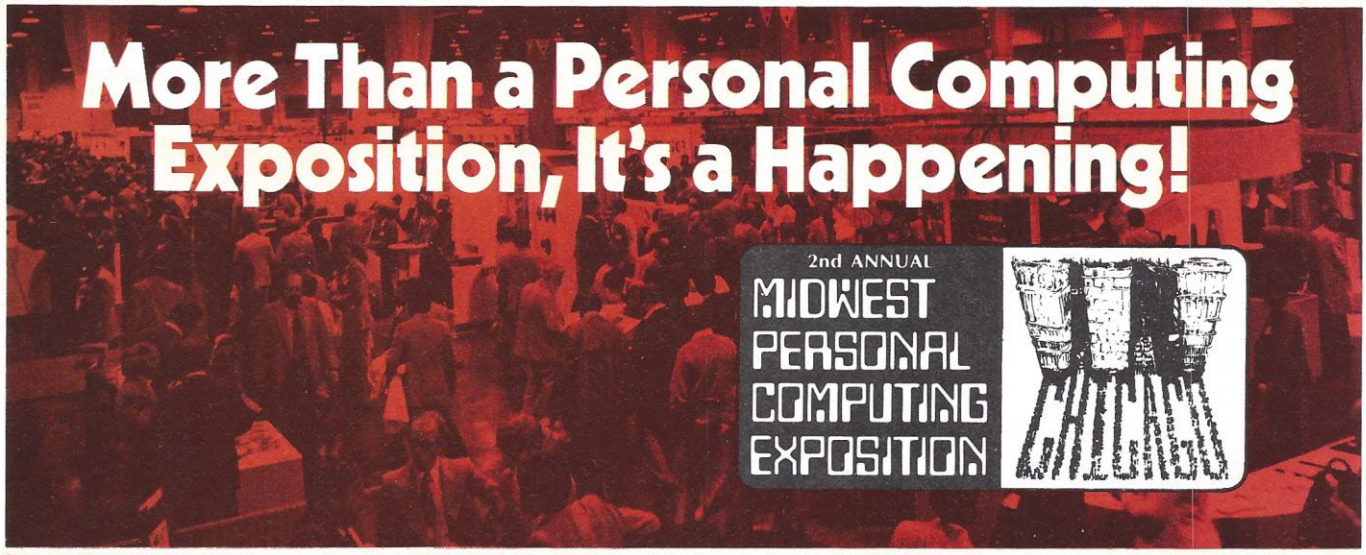


unit is installed between terminal and modem. Connect an X-Y recorder to **SLIP**, says the company, and presto! you have an economical and powerful plotting and graphics capability. User program statements may be in BASIC, FORTRAN, PL-1, COBOL or other popular languages. The user program need only output a list of integer coordinates to be plotted. More information and attractive price schedule may be obtained from the company. For more information, contact Special Systems, Inc. 8045 Newell Street, Silver Springs, MD 20910. Circle No. 106.

The MFE Corporation of Salem, NH, has a new line of buffered data cassette terminals – the **Model 2500.** The new unit uses MFE's 450 Tape Drive which allows recording on both sides of the tape for an un-



equalled 350,000 character capacity. ANSI compatible, the MFE 2500 is available with TI or NCR compatability, selectable rates up to 2400 baud and a binary mode. Also standard are TTY and RS-232C interfaces. Offering most of the features of its MFE 5000 model, says the company, the Model 2500 has greater storage at a much reduced price: $1,190. For more information, contact MFE Corp., Keewaydin Drive, Salem, NH 03079. Circle No. 103.



OCTOBER 5 - 8, 1978 • EXPOCENTER/Chicago (Adjacent To Merchandise Mart)

A week-long revel of special events, celebrity attractions, adventures into the future, surprises and assorted delights . . . **Now officially proclaimed, "Midwest Personal Computing Week" by Chicago's Mayor Bilandic.**

And highlighted by the second annual MIDWEST PERSONAL COMPUTING EXPO, which brings together the latest innovations in equipment, peripherals, accessories and software by such leading manufacturers, distributors and publications as:

Bits 'n Bytes Computer Stores
Compute-A-Bit
Digital Enterprises
Microcomp
Olson Electronics
Summagraphics
Telpar, Inc.
Artec Electronics
Cherry Electrical Products Corp.
Information Systems Inc.
Apple Computer Co.
Jade, Inc.
Hobbyworld
Radio Shack
Nabih's Inc.
Control Data Corp.
Chicago Computer Stores, Inc.
The Milwaukee Computer Store
Quantum Computer Works
Tek-Aids, Inc.
itty bitty Machine Co.
Compucolor Corp.
The Digital Group
Processor Technology
AAA Chicago Computer Center
Computer Electronic Modes, Inc.
Quay Corp.
Personal Computing
Minicomputer News
Chicagoland Business Services and Equipment News
Electronics Journal
Computer Dealer
Digital Design
Electronic Packaging and Production

Listing Complete As of May 1, 1978 . . . Many More Top-Name Exhibitors by October.

Special Emphasis on Meeting the Computer Needs of Small and Medium-Sized Businesses

Over 200 product displays anticipated by show time. 12,565 doctors, lawyers, business and computer professionals, educators, students, home users and hobbyists came last year. More than 20,000 expected in '78.

Easy access from all public transportation and expressways to Chicago's close-in Expocenter. Ample parking available.

Expanded seminar program for everyone — beginning or experienced. Details practical applications of today's personal computing technology, including:

- Introduction to BASIC
- How to Program Your Own Computer
- How to Use Resources of Million Dollar Computer Systems with Personal Computers
- Home Applications of Personal Computers — Tax Records, Environmental Control, etc.
- Computer Graphics for Business or Entertainment
- Time-sharing with a Microcomputer
- Plus many more.

Plan Now to Participate as Visitor or Exhibitor

The ONLY Personal Computing Show in Chicago Officially Sponsored by PERSONAL COMPUTING Magazine

Organized in 1978 by:

Industrial and Scientific Conference Management, Inc.
a respected leader in expositions and conferences, worldwide
222 W. Adams St., Chicago, IL 60606
(312) 263-4866

Return for Complete Information to:

MIDWEST PERSONAL COMPUTING EXPOSITION
c/o Industrial and Scientific Conference Management, Inc.
222 West Adams St., Chicago, IL 60606 • (312) 263-4866

☐ I am interested in ATTENDING. Please send registration information.

☐ I am interested in EXHIBITING. Please send contract and full details.

Name _____ Title _____

Company _____ Phone _____

Address _____

City _____ State _____ Zip _____

CLASSIFIEDS

BLACKJACK

The ONLY BJ program that plays Vegas rules
PLAYBJ-a BASIC program simulated for:
***multiple deck play *insurance**
***doubling down *splitting**
Only $10 includes listing, documentation, modification instructions. FREE DETAILS: Write Gambling Times Magazine, Dept. C, 839 N. Highland, Hollywood, CA 90038.

NEW!

Shugart Minifloppy SA400 . . .$285
Shugart Floppy SA800/1 . . . .$450
IBM Comp. 8" diskettes . . . . .$3.50 ea.
Verbatim 5" diskettes . . . . . $3.50 ea.
Save 10-50% on new and used computer equipment
COMPUTER INTERFACE TECHNOLOGY
(714) 979-9920
Specializing in DEC, DG, Interdata

PET OWNERS

Now, a great new idea in clubs! Join, and you will receive "PROGRAM," a cassette, each month.

You load — nothing to type — and it gives you articles, applications, and information . . . right on your screen. What's more, it also contains 3-5 games or programs useful and fun for your sweet PET. Membership is $27 a year. Send check or money order to:

PROGRAM
P.O. Box 461
Philipsburg, PA 16866

PERIPHERALS, SUPPLIES & SYSTEMS

8" IBM comp. DISK 3740, S/32	3.95
8", 32 "HARD SECTOR" DISK	3.95
5¼" 1,10, 16 SECTOR DISK	3.75
TELETYPE MOD 43, KSR/TTL	1000.00
TELETYPE MOD 43, RO/TTL	900.00
TELETYPE MOD 43, MODEM	1350.00
INTERTEC VIDEO TERMINAL	784.00
COMPLETE SM. BUSINESS SYSTEMS	6000.00
COMPLETE EDUCATOR SYSTEM	5600.00

N.Y. residents add sales tax
S&D COMPUTER TECHNOLOGY
PO Box 1283, Albany, NY 12201
CALL 24 HOURS/7 DAYS
(518) 463-2091

TIS

Documentation and software applications packages for the COMMODORE PET 2001. Workbooks from $3.95; software from $4.95.
For a flyer describing all our products, please send a self-addressed stamped envelope to TIS, P.O. Box 921. Los Alamos, NM 87544.

PET OWNERS

Ever look at Radio Shack's 232 pg. owner's manual and wish Commodore had one?

We have just published "PETABLE," an introduction to PET Basic, graphics, and capabilities. It includes programs, exercises software/hardware sources, a periodical index, and much more. Send check or money order for $4.95 to:

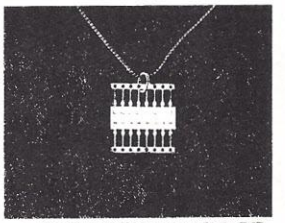
PETABLE
P.O. Box 461
Philipsburg, PA 16866

TRS-80 OWNERS!! Original software including business system for AMWAY Distributors. Level I & II versions available. All will run in 4K. Also have moving signboard machine language program. Write for info. Allan E. Sitter, 8 Driftwood Circle, Groton, CT 06340.

TRS-80 MONTHLY NEWSLETTER . . .

Send for details:
HOWARD Y. GOSMAN
Box 149, New City, New York 10956

Chips Ahoy!



Our First Offering is a 16-pin DIP IC with uncut gold leads and a white ceramic body in flat form (contains no IC circuit). This is looped and strung on a choice serpentine 14K **solid gold** chain 18" in length. **w/chain $45. charm alone 10.**

Send your order with cash, check, money order, BA/Visa, or Master Charge.

CHIPS AHOY!
POBox 1199
Portsmouth, NH 03801

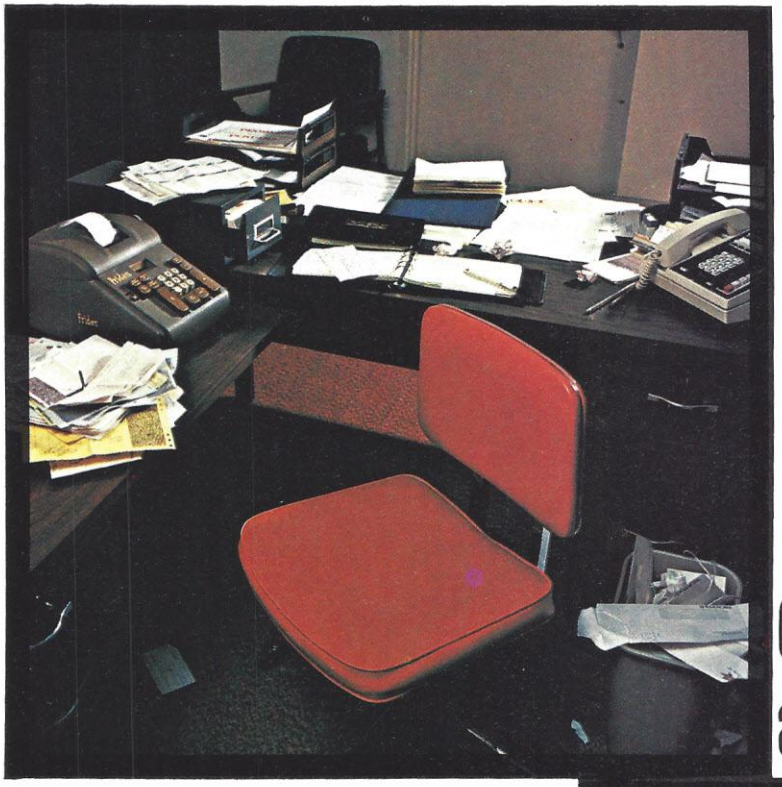
See us at Personal Computing '78 in Philadelphia, Aug. 24-27.

ADVERTISERS' INDEX

SALES OFFICES

Northeast: George Palken, 1050 Commonwealth Ave., Boston, MA 02215; (617) 232-5470 • **Mid-Atlantic**: Arthur Daks, 104 So. Central Ave., Valley Stream, NY 11580; (516) 872-9550 • **Midwest**: Hank Bean, 2633 Hillside Lane, Evanston, IL 60201; (312) 475-7173 • **Los Angeles**: Carol Stagg, 924 Westwood Blvd., Suite 610, Los Angeles, CA 90024; (213) 487-3017 • **San Francisco**: Ralph Petersen, 1180 Oak Creek Dr., Palo Alto, CA 94304; (415) 328-4392 • Japan: Hiro H. Irie, International Business Corp., 11-8, Narita-Higashi 1-chome, Suginami-KU Tokyo 166; Phone: (03) 311-1746 • Exposition Service Sales Office: Louise Garcia, (505) 898-4562.

Is your TRASH FLOW higher than your CASH FLOW?



SOFTWARE LIBRARY

Five diskettes are included to give you immediate operating and programming capabilities.

DISK 1 is a master diskette with BASIC, MDOS, Text Editor, Assembler and more!

DISK 2 has many games including LUNAR, CRAPS, and SPLAT. There's room left for you to add your own.

DISK 3 contains a Small Business Accounting package which includes Accounts Payable and Receivable, Inventory, a General Ledger, and more.

DISKS 4 & 5 are blank so you may add your own programs.

Clean it up with a VERSATILE 3B™

If you're a businessman, we know how difficult it is to keep neat and efficient accounting records. Let our **VERSATILE** systems do it for you. You'll have a complete system built into a single cabinet, and a free software library on diskette to get your computer working for you the first day. $3295 Assembled and tested.

WHAT ELSE DOES A COMPLETE SYSTEM NEED?

Contact Our Distributors for Regional Sales and Service

Alexander & Co., Inc.
5518 Florin Road
Sacramento, CA 95823
(916) 422-9070

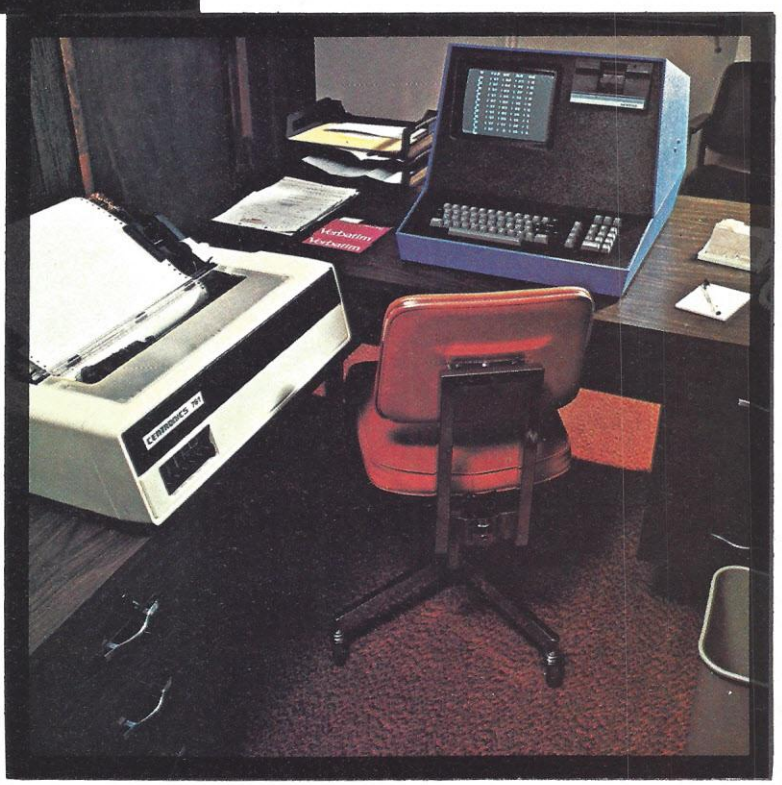
Huron Electronics
415 N. Silver
Bad Axe, Mich. 48413
(517) 269-9267

The Computer Store
3801 Kirby Dr., Suite 432
Houston, Texas 77098
(713) 522-7845

Southeast Representative:

Scientific Sales Co.
175 W. Wieuca, Suite 210
Atlanta, GA 30342
(404) 252-6808

DealerApplications Available.



COMPUTER DATA SYSTEMS

5460 Fairmont Drive • Wilmington, Delaware 19808 • 302-738-0933

CIRCLE 2

COSMAC VIP

$249* gets the entire family into creating video games, graphics and control functions. For starters.

COSMAC VIP, the completely assembled, ready-to-operate RCA Video Interface Processor, opens up a whole new world of computer excitement. New challenges in graphics, games and control functions. Yet it's just $249.00.

Easy to buy. And easy to program, thanks to its unique, easy-to-use interpretive language. You get a complete how-to book including programs for 20 games: fun, challenging, and ready to load and record on your cassette.

Simple but powerful.

Built around an RCA COSMAC microprocessor, the VIP is a complete computer system that can grow with you. It has 2K of RAM, expandable on-board to 4K. Plus a ROM monitor, audio tone output to a built-in speaker, power supply, and 8-bit input and output ports for control of relays, sensors, or other peripherals.

Soon RCA will offer options for color graphics and 256 tone sound generation. An optional auxiliary keyboard will open up an exciting world of two-player games.

Take the first step now.

Check your local computer store or electronics distributor for the VIP. Or contact RCA VIP Marketing, New Holland Avenue, Lancaster, PA 17604. Phone (717) 291-5848.

*Suggested retail price. Does not include video monitor or cassette recorder.

The fun way into computers.

RCA

CIRCLE 3